

魏克新 王云亮 陈志敏 编著

MATLAB语言与自动控制系统设计

电气自动化新技术丛书

机械工业出版社

电气自动化新技术丛书

MATLAB语言
与自动控制系统设计

魏克新　王云亮　陈志敏　编著

机 械 工 业 出 版 社

这是一本介绍MATLAB软件在自动控制系统分析、设计问题和仿真应用方面的专业书。全书共分10章，第1~3章介绍了与控制系统设计有关的MATLAB命令和SIMULINK仿真软件；第4~9章结合自动控制系统设计问题，介绍了应用MATLAB软件程序对自动控制系统进行分析和设计的方法，如状态空间设计方法、数字控制和线性二次型最优控制设计方法等；第10章介绍了MATLAB的其它软件工具箱。书中用大量的例题说明该软件在设计工作中的用法，并在每一章后给出了一定量的习题，以利于读者自学。

本书适用于从事自动控制设计的工程技术人员阅读，也可供大专院校有关专业师生参考。

图书在版编目（CIP）数据

MATLAB语言与自动控制系统设计/魏克新等编著-北京：机械工业出版社，1997.8
(电气自动化新技术丛书)
ISBN 7-111-05644-2

I.M… II.魏… III.①命令语言，MATLAB②自动控制系统-设计 IV. TP273

中国版本图书馆CIP数据核字（97）第11506号

出版人：马九荣（北京市百万庄南街1号 邮政编码 100037）
责任编辑：孙流芳 版式设计：张世琴 责任校对：林去菲
封面设计：姚 毅 责任印制：卢子祥
北京市密云县印刷厂印刷・新华书店北京发行所发行
1997年8月第1版第1次印刷
850mm×1168mm 1/32・13.875印张・ 360千字
0001-4500册
定价：25.00元

前 言

以往人们在自动控制系统设计过程中都使用笔算解决问题，运算量很大。尽管人们可借助于计算器的帮助，而对于大多数初学者来说，很少有人能够实际地计算出如阶跃响应，且描绘出响应的曲线波形。而且我们能够求出解析解的控制系统设计问题只限于二阶和三阶系统。后来，人们使用计算机来求解控制系统的设计问题。然而许多人花费了很多的时间，编制与调试程序，却都很难得到满意的结果。特别是对于初学者而言，事情似乎变得更糟。在许多情况下，其他人难以判断你编制的程序正确性与系统响应结果的正确程度。

MATLAB软件的诞生，使控制系统的分析与设计问题变得简单了。它为控制系统的设计与仿真提供了一个强有力的工具，必将使传统的方法产生一种变革。由于该软件具有容易使用、矩阵运算功能强和丰富的控制理论与CAD应用程序集（或称工具箱）等特点，已经成为国际控制领域内最流行的控制系统的计算机辅助设计软件。该软件的推广应用必将提高我们的控制系统的设计水平。目前我们国内已有不少研究单位、高等院校正在其控制系统的研究工作中使用或开始使用这种软件。许多专业科技人员、教师、研究生与大学生都需要一本如何使用该语言进行控制系统设计的专业书籍。为此，我们总结了近几年来国外与国内研究工作的经验，参考了一些有关资料而编写了这本书。

本书结合自动控制系统设计问题介绍了MATLAB软件及SIMULINK仿真软件的一般使用方法。该书分为两大部分。第一部分介绍了MATLAB常用的命令、控制系统工具箱及SIMULINK仿真工具软件。在这一部分中我们针对控制系统设计问题介绍了与其相关的MATLAB命令，并举例加以说明。在第二部分中，介绍了MATLAB软件程序的分析和设计方法，如状态空间设计方

法、数字控制、线性二次型控制和控制系统设计举例等。书中还介绍了MATLAB的一些软件工具箱。总之书中包括和涉及到了有关MATLAB各个方面的指令，覆盖了在控制方面较广的论题。书中用大量的例题说明该软件在设计工作中的用法，并在每一章后面给出了一定量的习题，以利于读者自学。它既是一本介绍MATLAB软件在控制系统分析、设计与仿真应用方面的专业书，又可作为大专院校学生与研究生的教材。我们希望该书的出版能够对从事控制系统研究的专业技术人员和专业教师的工作，以及对研究生和大学生的学习，给予一定的帮助。本书共有10章，第1～4章由魏克新副教授编写；第5～7、9章由王云亮副教授编写；第8、10章由陈志敏老师编写。在编写过程中，我们得到了丛书编委会和喻士林教授级高级工程师、天津理工学院岳有军老师和高强老师的大力支持，在此我们表示衷心的感谢。由于我们学识水平有限，书中不免有不妥与错误之处，希望能够得到各高校和各研究单位的专家教授、科技人员、教师和学员们的批评帮助。

作者

1997年5月

《电气自动化新技术丛书》

出版基金资助单位

机械工业部天津电气传动设计研究所
深圳华能电子有限公司
北京电力电子新技术研究开发中心
天津普辰电子工程有限公司

《电气自动化新技术丛书》

编辑委员会成员

《电气自动化新技术丛书》

序　　言

科学技术的发展，对于改变社会的生产面貌,推动人类文明向前发展，具有极其重要的意义。电气自动化技术是多种学科的交叉综合，特别在电力电子、微电子及计算机技术迅速发展的今天，电气自动化技术更是日新月异。毫无疑问，电气自动化技术必将在建设“四化”、提高国民经济水平中发挥重要的作用。

为了帮助在经济建设第一线工作的工程技术人员能够及时熟悉和掌握电气自动化领域中的新技术，中国自动化学会电气自动化专业委员会和中国电工技术学会电控系统与装置专业委员会联合成立了《电气自动化新技术丛书》编辑委员会，负责组织编辑《电气自动化新技术丛书》。丛书将由机械工业出版社出版。

本丛书有如下特色：

一、本丛书是专题论著，选题内容新颖，反映电气自动化新技术的成就和应用经验，适合我国经济建设急需。

二、理论联系实际，重点在于指导如何正确运用理论解决实际问题。

三、内容深入浅出，条理清晰，语言通俗，文笔流畅，便于自学。

本丛书以工程技术人员为主要读者，也可供科研人员及大专院校师生参考。

编写出版《电气自动化新技术丛书》，对于我们是一种尝试，难免存在不少问题和缺点，希广大读者给予支持和帮助，并欢迎大家批评与指正。

《电气自动化新技术丛书》
编辑委员会

目　录

第1章　MATLAB语言简介

MATLAB是一个高级的数学分析与运算软件，可以用作动态系统的建模与仿真。正如其名(矩阵实验室)，它非常适用于矩阵的分析与运算。MATLAB是一个开放的环境，在这个环境下，人们开发了许多具有特殊用途的工具箱软件，如控制系统、信号处理、最优控制、鲁棒控制及模糊控制工具箱等。本章主要介绍MATLAB文件管理、数据结构、数学运算、多项式及MATLAB绘图的一些规则及使用方法。MATLAB函数命令说明请见本章附录。

1.1　屏幕帮助

在MATLAB命令提示符下,使用help命令可以得到有关MATLAB命令的屏幕帮助信息。因此可以省去因记忆MATLAB命令而带来的麻烦。

输入

```
》help
```

就会在屏幕中给出下列所示的MATLAB的基本命令及与其相关的工具箱软件命令。

```
HELP topics:
toolbox\local      -Local function library.
MATLAB\datafun     -Data analysis and Fourier transform functions.
MATLAB\elfun       -Elementary math functions.
MATLAB\elmat       -Elementary matrices and matrix manipulation.
MATLAB\funfun      -Function functions-nonlinear numerical methods.
MATLAB\general     -General purpose commands.
```

```
MATLAB\color       -Color control and lighting model functions.
MATLAB\graphics    -General purpose graphics functions.
MATLAB\iofun       -Low-level file I/O functions.
MATLAB\lang        -Language constructs and debugging.
MATLAB\matfun      -Matrix functions-numerical linear algebra.
MATLAB\ops         -Operators and special characters.
MATLAB\plotxy      -Two dimensional graphics.
MATLAB\plotxyz     -Three dimensional graphics.
MATLAB\polyfun     -Polynomial and interpolation functions.
MATLAB\sounds      -Sound processing functions.
MATLAB\sparfun     -Sparse matrix functions.
MATLAB\specfun     -Specialized math functions.
MATLAB\specmat     -Specialized matrices.
MATLAB\strfun      -Character string functions.
MATLAB\dde         -DDE Toolbox.
MATLAB\demos       -The MATLAB Expo and other demonstrations.
SIM~\SIMULINK      -SIMULINK model analysis and construction
                   functions.
SIM~\simdemos      -SIMULINK demonstrations and samples.
SIMULINK\blocks    -SIMULINK block library.
SIMULINK\sb2sl     -System Build 3.0 model import into SIMULINK.
Toolbox\control    -Control System Toolbox.
Toolbox\ncd        -Nonlinear Control Design Toolbox.
Toolbox\ident      -System Identification Toolbox.
Toolbox\uitools    -User Interface Utilities.
Fuzzy\fuzzy        -Fuzzy Logic Toolbox.
Fuzzy\fuzdemos     -Fuzzy Logic Toolbox Demos.
Toolbox\wintools   -GUI tools for MATLAB for MS Windows.
For more help on directory/topic, type "help topic".
```

输入

》help command name

就可以得到这个命令的具体信息。例如输入

》help bode

就可以在屏幕上得到如下有关该命令的使用说明。

```
BODE  Bode frequency response for continuous-time linear systems.
BODE(A,B,C,D,IU) produces a Bode plot from the single input IU to
all the outputs of the continuous state-space system (A,B,C,D). IU
is an index into the inputs of the system and specifies which
input to use for the Bode response.The frequency range and number
of points are chosen automatically.
BODE(NUM,DEN) produces the Bode plot for the polynomial transfer
function G(s) = NUM(s)/DEN(s) where NUM and DEN contain the
polynomial coefficients in descending powers of s.
BODE(A,B,C,D,IU,W) or BODE(NUM,DEN,W) uses the user-supplied
frequency vector W which must contain the frequencies, in
radians/sec, at which the Bode response is to be evaluated. See
LOGSPACE to generate logarithmically spaced frequency vectors.
When invoked with left hand arguments,
        [MAG,PHASE,W] = BODE(A,B,C,D,...)
        [MAG,PHASE,W] = BODE(NUM,DEN,...)
returns the frequency vector W and matrices MAG and PHASE (in
degrees) with as many columns as outputs and length(W) rows. No
plot is drawn on the screen.

See also FBODE, LOGSPACE, SEMILOGX, MARGIN, NICHOLS, and NYQUIST.
```

一般MATLAB命令格式如下:

[输出参数1，输出参数2，...]=命令名(输入参数1，输入参数2，...)

输出参数使用方括号，而输入参数使用圆括号。如果输出量

仅一个，可不使用括号。

我们可以使用demo命令给出的范例去了解MATLAB的一般情况。输入命令后，MATLAB将显示如何使用MATLAB命令及各种命令的特点。范例具有很好的学习指导性，如果肯花上一些时间学习这些范例,你就会从中学到许多有关MATLAB的特点及其使用方法。建议在使用MATLAB之前，首先使用demo命令学习MATLAB的范例。这将有助于正确地使用MATLAB软件。

1.2 文件管理

在文件管理中，diary是一个很有用的命令。使用该命令可以在MATLAB命令窗口中以ASCII码形式记录所有的输入与输出。注意，这个命令不能存储数据，而可存储输入的命令与屏幕上输出的内容。它可以用来记录整个工作过程。因此，建议在每一部分工作开始之前使用diary命令，以便记录下工作过程，作为后续工作的备用文件。可以使用不同的文件名来区分所做工作的不同内容。否则，如使用相同的文件名将会使上一次存入的内容丢失。

输入

》diary filename

该命令即可打开一个以filename为文件名的文件。当然该文件名中可以包含路径。当完成工作后，使用

》diary off　　% close the diary

如只输入“diary”命令，则打开一个以“diary”为文件名的文件。工作完毕后，再输入“diary”，则以“diary”为名存储工作过程。使用%符号可以在程序中加入注释，即

》% 此处输入任何内容都将被忽略，不作为命令执行。

使用save命令，可以把数据以二进制的形式存储到一个文件中，如输入

》save filename

如果文件名中无扩展名，MATLAB自动添加.mat的扩展名。

该命令存储定义的变量或运算的结果，而diary命令只存储输入输出过程，不是实际的数据。save命令也可以用来存储指定的变量，如

》save filename X Y Z（存变量X，Y，Z到文件中）

输入

》save filename X/ascii/double

就会以ASCII码形式、双精度(16位)存X到以filename为名的文件中。使用该命令可以将MATLAB环境下的数据转换到其它种类的程序中去。

输入

》load filename

用一个给定的文件名装载数据文件。

输入

》what

在当前的目录下显示MATLAB文件与mat数据文件。

输入

》dir

显示在当前目录下的所有文件。与DOS命令相同，在MATLAB环境中可以使用delete命令删除文件；使用chdir命令改变当前目录；也可以使用type命令观看ASCII码文件内容。

输入

》clear a b c

从工作空间中清除指定的a、b、c变量。如果仅仅输入clear命令，则清除整个工作空间。

输入

》!

感叹号命令使你从MATLAB工作空间转移到操作系统环境。如果计算机有足够的存储空间，还可以装入其它程序，如EDITOR等。请注意，该命令不适用于在Windows环境中使用（因为也没有必要在Windows环境中使用它）。

在MATLAB环境中有两种修正输入错误的方法。其一，如输入数据后，还没有按回车键，此时可以使用退后键（←)来修改已输入的错误；其二，也是经常使用的方法，即使用上移键（↑）或下移键（↓）来重现已输入的数据或命令，然后修改。

在MATLAB环境中，大写与小写字母表示的意义是不同的，如A与a不代表同样的变量。所有的MATLAB命令要使用小写字母。

输入

》exit　　或者　》quit

执行以上命令可以退出MATLAB环境。

1.3　数据结构：矢量与矩阵

MATLAB的基本元素是双精度的复数矩阵。这是它的一般表达方法，而且也包含了实数和复数矢量与常数。它也间接地包含了多项式与传递函数。在MATLAB环境下，输入一行矢量很简单，只需使用方括号，并且每个元素之间用空格或用逗号隔开即可。列矢量的输入只需在行矢量输入格式基础上再加一个转置符号（'）即可。例如:

```
》x=[1,2,3],y=[1+j,2+pi*i,-sprt(-1)]
x=
   1  2  3
y=
   1.0000+1.0000i   2.0000+3.1416i   0-1.0000i
```

由上可见，可以在单一的命令行中输入几个矢量。它们之间要用逗号或分号隔开。

一些常用的数有特定的名字。如pi(=3.1416…)，j或者$i(=\sqrt{-1})$。其它的特殊函数还有Inf（∞；有限数被零除的结果）；NaN表示由一些不确定的形式所求得的结果，如0/0。当进行如0/0的运算时，会从屏幕上得到一个警示，而不是一个出错信息。你仍然可以给它们定义其它的数值，或者用其它名字表示这些特

殊函数。注意，当定义一个矢量中的元素时，可以使用任何数学函数或表示方法。例如，在上述定义矢量y时，可以使用sqrt函数来替代$\sqrt{-1}$。在下例中首先定义矢量z，然后求其转置。

```
》z=[1+j，2+pi*i，-sqrt(-1)]'
z=
  1.0000-1.0000i
  2.0000-3.1416i
  0 + 1.0000i
》z.'
ans=
        1.0000+1.0000i 2.0000-3.1416i 0+1.0000i
```

当对复数矢量进行转置操作时，你可以得到其共轭转置矢量。从每一元素虚部符号变化可以看出，它与对应元素是共轭的。如上面所示，如果在z的后面加一点，然后再加转置符号，则可以得到非共轭转置矢量。在上述操作中，没有给出z'的变量名，而MATLAB自动使用恒定变量ans作为变量名。恒定变量与其它任何变量相同，只是它不能被从MATLAB工作空间清除。恒定变量命令还有eps，它表示最小的机器数。

输入

```
》eps
eps=
        2.2204e-016
```

即可得到最小的机器数。

在MATLAB中，冒号“：”是很有用的命令符。例如：

```
》t=[0:0.1:10]；
```

它产生一个从0到10的行矢量，而且元素之间间隔为0.1。如果增量为负值，可以得到一个递减的顺序矢量。使用linspace与logspace命令也可以得到这样顺序排列的矢量。例如：

```
》t=linspace(n1,n2,n)
》w=logspace(n1,n2,n)
```

linspace命令可以在线性空间产生一个行矢量，行矢量的值从n_1到n_2，在n_1与n_2之间的点数为n。如不输入参数n，缺省值为100。logspace命令可以在对数空间产生一个行矢量,其值从10^{n1}到10^{n2}，在它们之间的点数为n。如不输入参数n，则缺省值为50。该命令为在计算时域与频域响应中建立时间轴与对数频域轴坐标提供了方便。例如:

logspace(−1,2)　从0.1到100产生50个点；

logspace(0, pi)　从1到π产生50个点。

如果不希望立即在屏幕上显示数据，则在每个命令行后面加入分号即可。

如果没有在命令行末尾加入分号，则变量的数据或运算的结果将在屏幕上显示出来。

较长的变量名或命令需要多于一行的空间输入数据。可在当前的末端输入

》...

就可以连续输入命令。

矩阵的输入需要逐行输入，每个行矢量之间要用分号隔开或者回车。例如对于下列矩阵：

$$A=\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix}$$

输入

》A = [1 2 3; 4 5 6; 7 8 9]

每个数据之间的空格数可以任意设定。

who、whos、size和length是对提供工作空间变量信息很有用处的四个命令。各命令分别说明如下：

输入

》who

执行该命令可列出存储空间中的所有变量；

输入

》whos

显示所有的变量、变量所占的字节数及该变量是否是实数；

输入

》size(a)

执行该命令可以得到矢量a的行数与列数；

输入

》length(a)

执行该命令后，在屏幕上显示出矢量a的长度。如果a是矩阵，则显示的参数为行数与列数中的最大数。以上这四个命令在矩阵代数运算中是非常有用的。

显示的数据形式可以由数据格式化命令来修改。例如：

输入

》format long

则变量以15个数字定点格式显示。

1.3.1 矩阵的标号

由于矩阵在MATLAB中是一基本数据形式，因此要求使用者对矩阵元素的定位很熟悉。此外，MATLAB能够对矩阵进行各种运算，这在其它语言中是难以做到的，其中有些运算在线性代数中并没有定义，而有一些甚至是不合法的。矩阵元素定位地址方式如下：

$A(m, n)$

其中，m为行号，n为列号。例如：$A(3,4)$表示第三行第四列元素；A（：，2）表示所有的第二列元素；A（1:2,1:3）表示从第一行到第二行和第一列到第三列的所有元素。请注意此处“：”符号的用法。

如果在原矩阵中一个不存在的地址位置设定一个数，则该矩阵自动扩展行列数，并在该位置上添加这个数，而在其它没有指定的位置补0。例如原矩阵为

```
》a=[11,12; 21,22]
  a=
       11   12
       21   22
```

若在第3行，第3列添加一个数33，则输入

```
》a=（3，3）=33
  a=
       11   12   0
       21   22   0
       0    0    33
```

矩阵的下标不一定为整数，即使为小数，MATLAB也是可以接受的。例如，***A***（2.2，3）被认为是第二行第三列。但是矩阵的下标必须为正数。输入***A***（：）可以得到一个长的列矢量，该矢量的元素是按列一一叠加在一起的。例如：

```
》a=[1 2; 3 4]; a(:)
 ans=
       1
       2
       3
       4
```

矩阵的下标也可以是矢量。例如：

```
》b=a（x，y）
```

可以得到一个矩阵***b***。***a***的行标来自矢量***x***，而列标来自矢量***y***。

例如，矩阵***a***有***n***列，那么

```
》b=a（:，n:-1:1）
```

将使矩阵***a***按列的逆序排列。即矩阵***b***的第一列为矩阵***a***的第***n***列；矩阵***b***的第二列为矩阵***a***的*n*－1列；...。这个特点是十分有用的，可以使矩阵变换变得简单了。

1.3.2 特殊矩阵

一些常用的矩阵可作为一函数给出。使用eye命令可以得到一个单位矩阵。例如：

》eye(m)

就可以得到$m \times m$的单位矩阵；再如：

》eye(a)

可得到与矩阵a同样大小的单位矩阵；输入eye（m，n）命令可得到一个可允许最大的单位矩阵而其余处补0。

输入

》one(n)

可得到一个$n \times n$的“1”矩阵；输入ones（m，n）命令可以得到一个$m \times n$的“1”矩阵；而输入ones(a)命令可以得到一个与矩阵a相同行列数的“1”矩阵。这些函数在矩阵运算中是十分有用的。zeros命令与ones命令相似。所不同的是用“0”取代“1”。rand命令同样与上面的命令功能相似，所不同之处是rand命令得到的矩阵元素是在0和1之间的随机数。

另外一个特殊矩阵为空阵，这在线性代数中是不存在的。例如：

》q=[]

矩阵q在工作空间之中，但它的大小为零。通过空阵的办法可以删去矩阵的行与列。例如：

》a=rand（5，5）；a（:，1:3）=[]

第一条指令执行后得到一个5×5的矩阵；第二条指令将矩阵a的前三列删除。

另外一个特殊矩阵是对角矩阵。例如执行下列命令：

》diag(v)　　　% v为矢量

就可得到一个矢量元v在对角矩阵的对角线上。如果v是一个矩阵，执行该命令后得到一个列矢量，其元素为矩阵v对角线上的元素。

输入

》diag（v，1）

可得到矩阵v超对角线元素组成的列矢量。

输入

》diag（v，-1）

可得到矩阵v次对角线元素组成的列矢量。

其它相关的命令还有triu和tril，使用该两命令可以得到相应矩阵的上三角矩阵与下三角矩阵。这些命令的具体使用方法，可通过执行help命令去了解。

1.3.3 字符串

字符串要用单引号并用括号括在里面。例如：

```
》disp（ 'text string'）          % disp显示命令
  text string
```

在单引号里边的字符串可以作为矢量或矩阵的元素。使用disp命令或输入变量名就可以显示它们表示的字符串。例如:

```
》a=['this is a'; 'text string']
```

由屏幕上可以看到

All rows in the bracketed expression must have the same number of columns.

我们只要在第一个字符串中加入两个空格即可解决这个问题。

```
》aa=[ 'this is a  '; 'text string' ]
  aa=
      this is a
      text string
```

还有几个字符串分布命令可以用来作为文字说明和绘图标题说明等，如num2str、int2str、fprintf和sprintf。同样，可以借助于help命令了解它们的具体用法。

1.4 数学运算与函数

MATLAB可完成基本代数运算操作+、－、*、\、/、^（^表示平方）、标准三角函数、双曲线函数、超越函数(log为自然对数,log10为以10为底的对数)及开平方（√）等。

MATLAB还可以完成其它的矩阵函数运算，如求行列式(det)、矩阵求反(inv)、求矩阵特征值(eig)、求秩(rank)、求迹(trace)和模方(norm)等。强大的矩阵运算函数是MATLAB运算功能的核心。其它运算功能还有，求一个数的实部(real)、求一个数的虚部(imag)、求一个数的绝对值(abs)(复数的绝对值或幅值)和求共轭运算(conj)(复数共轭)。MATLAB还具有元素对元素的操作运算功能，它们对矩阵的数据处理是很有用的。但是这种运算在线性代数中是不合法的。可以使用help命令得到它的完整的运算函数功能表。

1.4.1 基本运算

MATLAB有两种数学运算函数：矩阵型与数列型（元素对元素）。由于数列型运算不属于线性代数中的标准线性代数，因此人们并不熟悉它的运算规则。然而，你会发现这种运算对数据分析是很有用的。

矩阵的加减和乘除是按照基本线性代数规则进行运算的。矩阵的幂运算在后面会给出定义。矩阵的求反(除法)由下式定义：

```
》 A\B
》 B/A
》 inv(A)
```

式中，A\B等于$A^{-1}B$或inv(A)*B（inv(A)为对矩阵A求逆)；B/A相当于BA^{-1}或B*inv(A)。

数列加、减、乘、除（./，.\）和幂(.^)运算都是元素对元素的运算。所有数列型运算要求参加运算的数列有相同的行列数。这些运算符前面要加一点。

A.*B是元素对元素的乘法，得到 $c_{ij} = a_{ij}\ b_{ij}$ 的矩阵。而***A./B***、***A.\B***和***A.^B***运算可以相应地得到a_{ij}/b_{ij}、b_{ij}/a_{ij}和a_{ij}^b_{ij} 的矩阵($i\,j$均为下标)。例如：

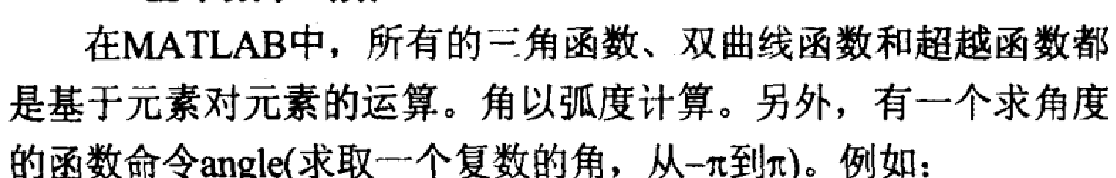
```
》 a=[1 2; 3 4];b=[5 6; 6 8];
》 a.*b
  ans=
```

```
     5    12
     18   32
》 a. \b
   ans=
     5    3
     2    2
》 a. ^b
   ans=
     1        64
     729      65536
```

通过输入以下命令可以证明sin(2*t*)=2sin(*t*)cos(*t*)的恒等性。

```
》 t=[0:100]'; x=sin(2*t); y=2*sin(t).*cos(t);
》 [x, y]
```

1.4.2 基本数学函数



在MATLAB中，所有的三角函数、双曲线函数和超越函数都是基于元素对元素的运算。角以弧度计算。另外，有一个求角度的函数命令angle(求取一个复数的角，从−π到π)。例如：

```
》 x=[1+j, -1-j]; angle(x)*180/pi
  ans=
      45   -135
```

用线性代数的方法，可定义矩阵函数(如果存在的话)如下：

$$f(\boldsymbol{A})=\boldsymbol{M}f(\Lambda)\boldsymbol{M}^{-1}$$

Λ是特征值矩阵，而M是矩阵的特征矢量矩阵。下例也是常用的矩阵函数。如：expm、logm、sqrtm和funm，funm函数可计算任何一个基本数学函数的矩阵函数。它可表示为

```
》 fa=funm (a, 'fun')
```

式中，fun可以是任意一个基本函数，如：sin，cos，$\log_{10}$等。

1.4.3 数据分析：列函数

列函数是面对矢量的运算。如果将它们用于矩阵，则它们是基于列的排列进行运算的。这些函数是：最小值(min)、最大值

(max)、平均值(mean)、中值(median)、标准误差(std)、求和(sum)、乘积(prod)、累积(cumsum)和累积乘(cumprod)。它们对应求矢量的最小值、最大值、平均值、中值、标准误差、求和、乘积、累积和累加乘积。对于矩阵的运算，所得结果为一行矢量。其值分别对应矩阵相应的列。为了求得一矩阵元素总和，可以将sum函数串联使用，即sum（sum（a））。同理，我们可以使用min和max函数求得矩阵中的最小数与最大数。例如：求阶跃响应中的峰值。

```
》[Mp，Jp]=max[ystep]
```

执行sort命令可使矩阵的每一列按升序排列。对于复数，它们的升序排列是按其幅值大小排列的。

样条内插spline命令用于三次的样条插值运算。它的命令格式为

```
》yi = spline(x，y，xi)
```

设$y=f(x)$，x对应在一段区间内。spline命令可以求得x点的y_i。下例说明了如何应用该命令。首先我们求得x在[1，5]区间的对数。然后再使用插值求得x=2.5的对数。下面同时也给出了对应x=2.5时的准确y值。

```
》x=[1:5]；y=log10(x)；
》y2=spline(x,y,2.5)
》y2=0.3993
》log10(2.5)
  ans=
      0.397
```

1.5 多项式

多项式表示以降阶排列含有多项式系数的矢量。利用求根（root）命令，可以求得多项式的根。

例如：求s^3+2s^2+3s+4的根可用如下命令：

```
》p=[1 2 3 4]；root(p)
```

```
ans=
        -1.6506
        -0.1747+1.5469i
        -0.1747-1.5469i
```

求多项式(poly)命令的功能是由多项式的根求得一多项式。其结果是由多项式系数组成的行矢量。其命令如下

```
》P2 = poly([-1 -2])
  P2 =
      1  3  2
```

如果poly的命令输入参数为矩阵，则可得到那个矩阵的特征多项式（行矢量）(特征多项式是A=det(λI–A))。

使用polyval命令可以计算在某一点多项式的值。它的格式为

```
》Ps = polyval (p, s)
```

式中，p是多项式，而s是要计算的点，s可以是一个矢量或一个矩阵，如果s是一个矩阵，其运算结果也是一个矩阵，且其元素与s的元素对应。例如：对多项式 $P(s)=(s+1)(s+2)$，可采用如下命令进行计算：

```
》P2 = [1 3 2]; a = [1 2; 3 4]; poly(P2, a)
  ans =
      6   12
      20  30
```

polyvalm是以矩阵的形式计算多项式的命令。例如：对

$$A=\begin{bmatrix}1 & 2\\3 & 4\end{bmatrix} \qquad p(A)=A^2+3A+2I$$

可采用如下命令进行计算：

```
》polyvalm (p, a)
  ans=
      12    16
      24    36
```

conv和deconv命令是多项式的乘与除命令。对应地

```
》c=conv (a, b)
》[q,r]=deconv (a, b)
```

实际上，conv命令是执行离散型卷积运算，等效于多项式的乘法；deconv执行的是长除法，其中q是商，r是余数。例如：

```
》conv ([1 1], [1 2])
  ans=
      1  3  2
》deconv([1 3 2],[1 1])
  ans=
      1  2
```

residue是部分分式命令。例如：

```
》[r, p, k]=residue (1, [1 3 2])
  r=
      -1
       1
  p=
      -2
      -1
  k=[]
```

输出r，p和k是部分分式系数、极点和常数系数。因为转换的函数恰好配整，所以k项是空阵。residue命令也可以用于从部分分式形式转换为分式多项式(num, den)形式。例如：

```
》[n, d]=residue(r, p, k)
  n=
      0  1
  d=
      1  3  2
```

MATLAB有几个命令用于微积分型问题的数值解，如数值积分命令quad、普通微分方程解命令ode23、寻找非线性函数零点命令fzero、函数最小值命令fmin和非线性函数方程系统解命令

fsolve。使用help命令可以得到有关使用这些命令的详细说明。

1.6 绘图命令

MATLAB具有很强的图形功能，非常适用于科学研究与工程实际中的应用。我们将举一些例子说明它的绘图功能与特点。绘图的基本命令有plot、loglog、semilogx、semilogy和polar。它们的使用方法基本相同，其不同点是在不同的坐标中绘制图形。plot命令使用线性坐标空间绘制图形；loglog命令在两个对数坐标空间中绘制图形；而semilogx(或semilogy)命令使用x轴(或y轴)为对数刻度，另外一个轴为线性刻度的空间绘制图形；polar命令使用极坐标空间绘制图形。例如：

》plot（x，y）

该命令是绘制y对应x的轨迹的命令。y与x均为矢量，且具有相同的元素数量。如果其中有一个参数为矩阵，则另一个矢量参数分别对应该矩阵的行或者列的元素可绘制出一簇曲线（究竟是对应行还是列绘制函数曲线，取决于哪个参数排在前面）。如果两个参数都是矩阵，则x的列对应y的列绘制出一簇曲线。

如果y是复数矢量，那么plot(y)将绘制该参数虚部与实部对应的曲线。该命令的这个特点在绘制奈魁斯特图时是很有用的。

如果在同一坐标内绘制多条曲线（对应某一坐标轴，具有相同的取值点）,可以由数据组成一个矩阵来同时绘制多条曲线。如下例共有三套数据，要求在同一坐标轴内同时绘制三条曲线。其命令格式如下：

》plot(t，[x1 x2 x3])

如果多重曲线对应不同的矢量绘制，可使用如下命令格式：

》plot（t1，x1，t2，x2，t3，x3）

式中表示x1对应t1，x2对应t2等等。在这种情况下，t1、t2和t3可以具有不同的元素数量，但要求x1、x2和x3必须分别与t1、t2和t3具有相同的元素数量。

subplot命令使得在一个屏幕上可以分开显示n个不同坐标，

且可分别在每一个坐标中绘制曲线。其命令格式如下：

》subplot（r c p）

该命令将屏幕分成r*c个窗口，而p表示在第几个窗口。例如：subplot（2，1，2），将屏幕分成两个窗口。subplot（2，1，1）与subplot（2，1，2）命令常用于控制系统伯德图(Bode)的绘制。窗口的排号是从左到右，自上而下。

在图形绘制完毕后，执行如下命令可以再在图中加入题目、标号、说明和分格线等。这些命令有title、xlabel、ylabel、gtext和text等。它们的命令格式如下：

》title（'My Title'），xlabel（'My X-axis Label'）

》ylabel('My Y-axis Label')

》gtext('Text for annotation')

》text(x，y，'Text for annotation')，grid

gtext命令是使用鼠标器定位的文字注释命令。当你输入命令后，可以在屏幕上得到一个光标，然后使用鼠标器控制它的位置。按鼠标器的左键，即可确定文字设定的位置。该命令使用起来非常方便。

shg和clg是显示与清除显示屏图形的命令。hold是图形保持命令，可以把当前图形保持在屏幕上不变，同时在这个坐标内绘制另外一个图形。hold命令是一个交替转换命令，即执行一次，转变一个状态（相当于hold on、hold off)。

MATLAB可以自动选择坐标轴的定标尺度，也可以使用axis命令定义坐标轴的特殊定标尺度。其命令格式如下：

》axis（[x-min，x-max，y-min，y-max]）

它可置坐标轴为特殊刻度。设置坐标轴以后，plot命令必须重新执行才能有效。axis命令的另一个作用是控制纵横尺度的比例。例如，输入axis（'square'）后，可得到一个显示方框。此时再在该框内绘制一个圆形时（如：plot（sin(x)，cos（x)），在屏幕上可以看到一个标准的圆（一般情况下，由于屏幕的不规则原因，只能看到一个椭圆）。再次输入axis（'normal'）命令，屏幕返回

到一般状态。

线型（-，—，:,-·）、符号（+，*，o，.，X）与线的颜色（r=red，g=green，b=blue，w=white，i=invisible）都可以被修正。有关三维绘图命令的使用方法请见下面的例题。

1.7 例题

【例1-1】绘制相对地球运动的水星运动轨迹。 其运动方程为

$$x(t)=93\cos t+36\cos 4.15t$$

$$y(t)=93\sin t+36\sin 4.15t$$

经过$7^1/_3$周期可以得到如下曲线。首先计算矢量x与矢量y，然后绘制运动轨迹曲线，MATLAB程序1-1为绘制运动轨迹的程序清单。其运动轨迹见图1-1。

```
% MATLAB PROGRAM 1-1
t=[0:pi/360:2*pi*22/3];
x=93*cos(t)+36*cos(t*4.15);
y=93*sin(t)+36*sin(t*4.15);
plot(y,x), grid,
```

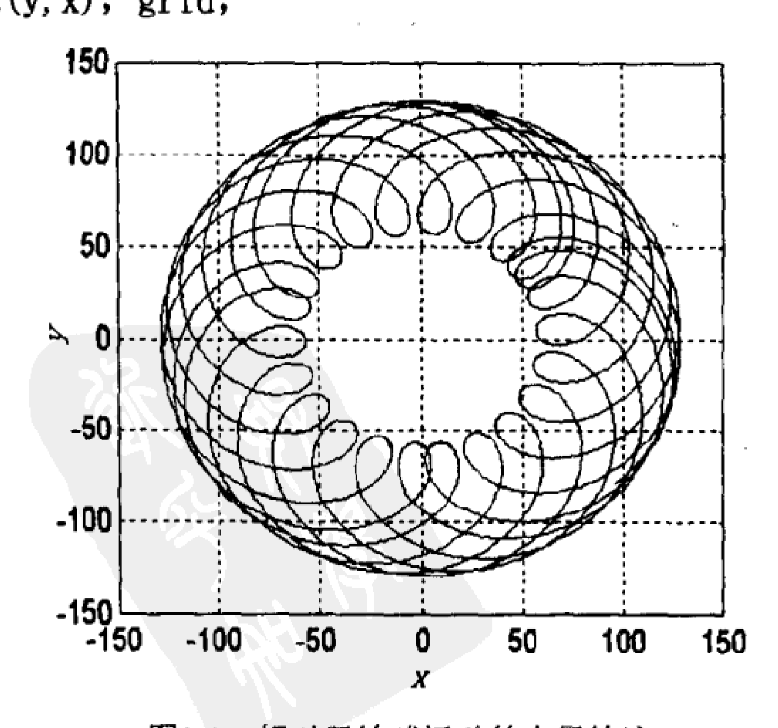


图1-1 相对于地球运动的水星轨迹

【**例1-2**】绘制极坐标图。

在极坐标系中的四叶方程是r=cos（2θ）。

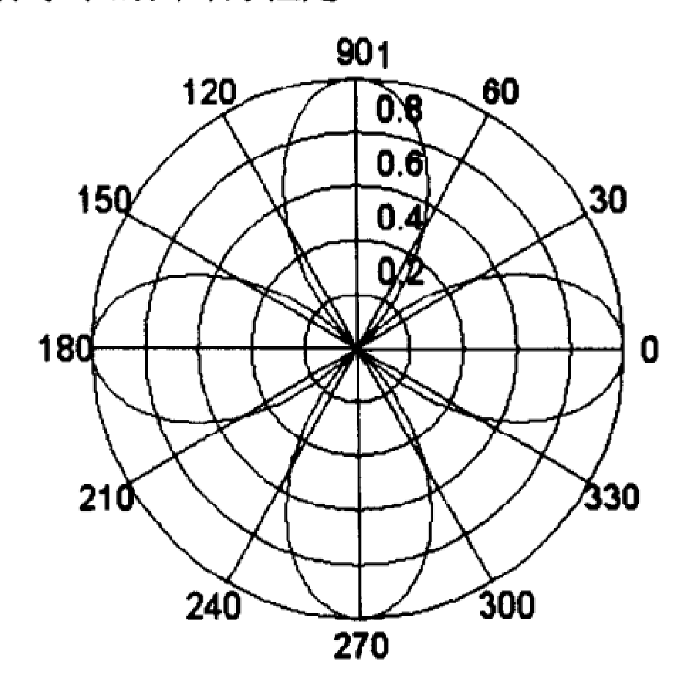


图1-2 四叶图形（极坐标）

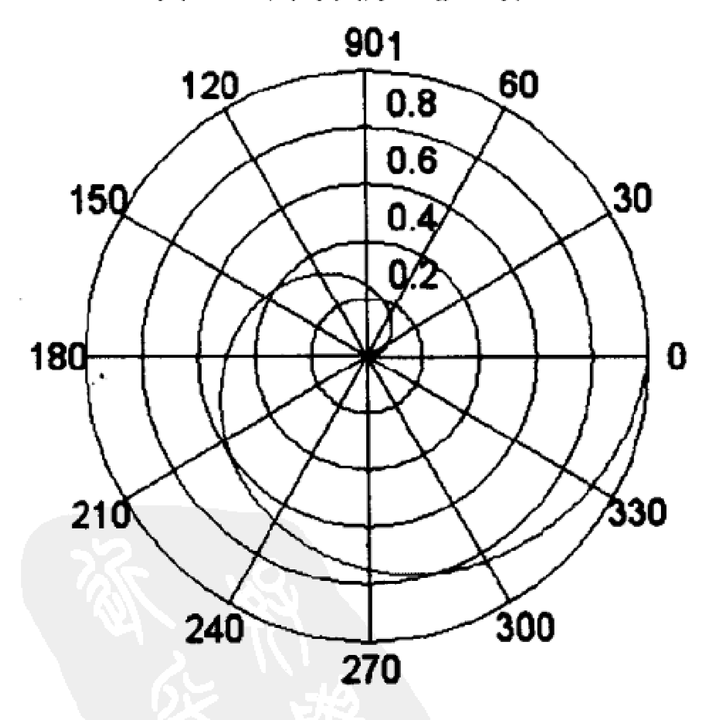


图1-3 阿基米德螺旋线

在极坐标命令中，角度用弧度表示。MATLAB程序1-2为绘制极坐标曲线的程序清单。极坐标曲线见图1-2。

```
% MATLAB PROGRAM 1-2
th=[pi/200:pi/200:2*pi]'; r=cos(2*th);
polar(th,r),grid
```

该方程的阿基米德螺旋线由$r = k\theta$（k>0）的方程给出，见图1-3。MATLAB程序1-3为绘制阿基米德螺旋线的程序清单。

```
% MATLAB PROGRAM 1-3
th=[pi/200:pi/200:2*pi]';
sa=th/(2*pi);
polar(th,sa), grid, pause
```

【例1-3】绘制圆曲线。绘制一个圆心在坐标原点、半径为r的圆，即$z = re^{j\theta}$。

MATLAB程序1-4为绘制同心圆的程序清单。图1-4显示了使用五种不同符号的点绘制的同心圆。

```
% MATLAB PROGRAM 1-4
th =[0:pi/10:2*pi]; x1=exp(j*th);
plot(real(x1), imag(x1), '.', ...
real(2*x1), imag(2*x1), 'o',...
real(3*x1), imag(3*x1), '+', ...
real(4*x1), imag(4*x1), '*',...
real(5*x1), imag(5*x1), 'x'), grid
```

请注意命令行的连续符号（...）的用法。由上述命令及图形可以看出，为区别不同的曲线可以采用不同的线型与符号。

对于半径为r、圆心为（a，b）坐标圆的参数方程为

$$x(t) = r\cos(t)+a \qquad y(t)=r\sin(t)+b$$

$$(x-a)^2+(y-b)^2=r^2$$

要绘制一个单位圆的圆心在坐标原点、另一个单位圆的圆心在（1，1）的两个圆，请见如下程序。MATLAB程序1-5为绘制不同圆心两个圆的程序清单。

```
% MATLAB PROGRAM 1-5
```

```
tt=[0: 0.1: 2*pi]'; x=cos(tt); y=sin(tt);
plot([x x+1], [y, y+1]), grid, axis( 'normal')
axis([-2 3 -2 3]);
```

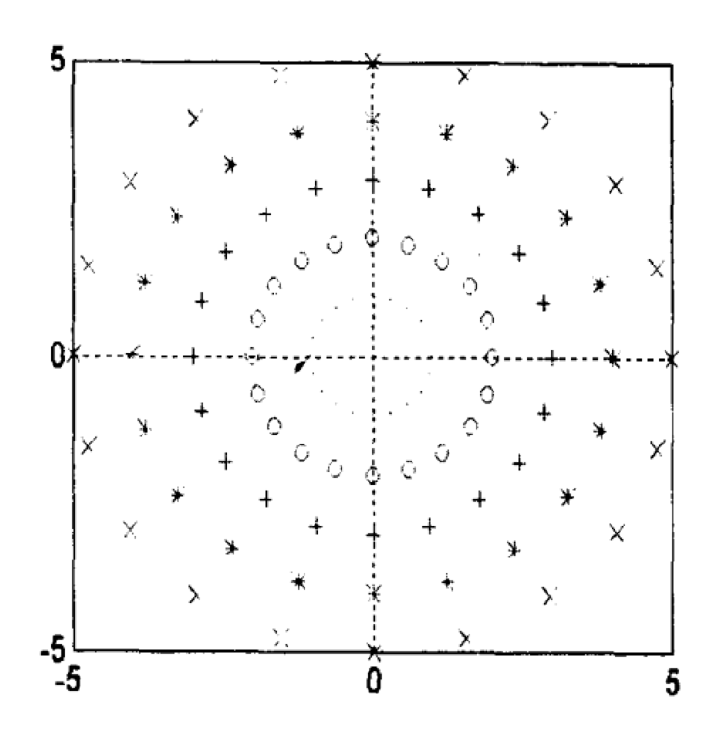


图1-4　五种不同符号的点绘制的同心圆

绘制的两个不同圆心的单位圆见图1-5。

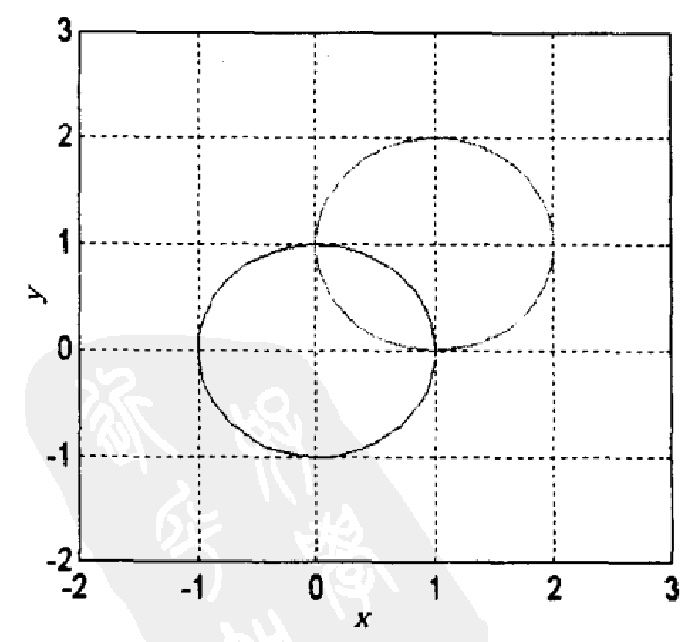


图1-5　两个不同心的单位圆

注意图形坐标轴的设置。也不要忘记将坐标设置复位，以备后续图形的绘制。

【例1-4】 绘制几何图形：三角形

在MATLAB环境下绘制线与几何图形是很方便的。只需要将x轴坐标作为一个矢量，而y轴坐标作为另外一个矢量，然后绘制y对x的曲线图形即可。为得到一个闭合的曲线图形（如三角型等），初始坐标点与最后的坐标点必须相同。利用下列程序，可以构造一个变换矩阵，通过变换可以把三角形的顶点旋转45°。MATLAB程序1-6为绘制三角形的程序清单。

```
% MATLAB PROGRAM 1-6
a=[1; 1]; b=[3; 1]; c=[2.5; 2];
% the vertices
tx=[a(1) b(1) c(1) a(1)]; % x coordinates
ty=[a(2) b(2) c(2) a(2)]; % y coordinates
aa=[cos(pi/4) -sin(pi/4); sin(pi/4) cos(pi/4)];
% rotation matrix
b1=aa*b; a1=aa*a; c1=aa*c; % new vertices
tx1=[a1(1) b1(1) c1(1) a1(1)];
% new x coordinates
ty1=[a1(2) b1(2) c1(2) a1(2)];
% new y coordinates
axis([0 3.5 1 3.5]),
plot(tx, ty), hold on, ...
plot(tx1, ty1), hold off, grid,
```

注意hold命令的用法，它可以使绘制在坐标中的第一个图形保持不变，再在该坐标中绘制第二个三角图形，见图1-6。

【例1-5】 三维图形的绘制。

mesh与meshdom命令用于两个变量的三维立体表面图形的绘制。如果*z*为矩阵，mesh（z）是以图形方式观察该矩阵的有效方法。如果我们要绘制$z=f(x, y)$的图形（其中*x*、*y*定义了*f*的区

间），首先我们定义矢量x与y的区间（注意：绘制三维图形使用的存储空间较大，时间也较长，因此要合理地选配定义空间，以保证不超出计算机的内存容量），然后需要求一个对应该区域的面。网状表面就构造在这个平面的顶部，其高度由对应的函数值大小来决定。meshdom命令通过构造两个矩阵产生这个平面，这两个矩阵是由重复构造矢量x与y的行与列构成的。最后在这两个矩阵元素基础上计算出其函数值。

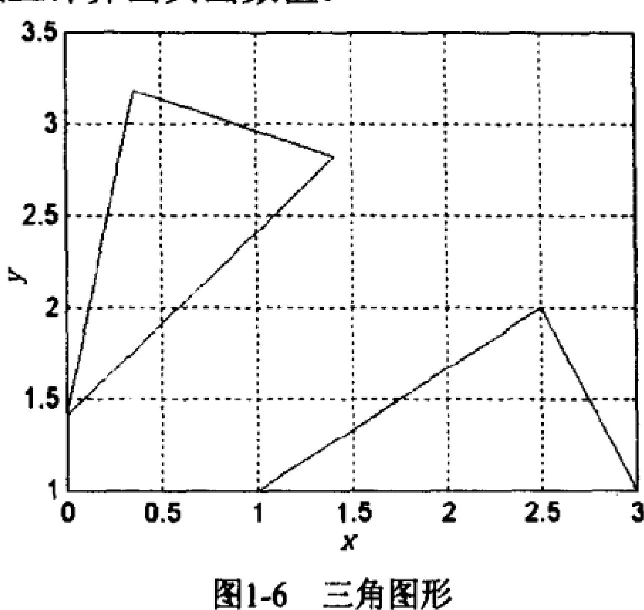


图1-6 三角图形

下面通过绘制二阶系统频率响应三维表面图来说明上述过程。二阶系统传递函数为

$$G(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

由$s=\sigma+j\omega$，可以得到

$$G(\sigma+j\omega)=\frac{\omega_n^2}{(\sigma^2-\omega^2+2\zeta\sigma\omega_n+\omega_n^2)+j\,(2\sigma\omega+2\zeta\omega\omega_n)}$$

对应ζ=0.5和ω_n=1，则

$$G(\sigma+j\omega)=\frac{1}{(\sigma^2-\omega^2+\sigma+1)+j(2\sigma\omega+\omega)}$$

该两变量函数的绘图区域为 $-3\leqslant\sigma\leqslant3$ 和$-3\leqslant\omega\leqslant3$ 。

MATLAB程序1-7为绘制二阶系统频率响应三维表面图的程

序清单。

```
% MATLAB PROGRAM 1-7
w=linspace(-3,3,50); s=linspace(-3,3,50);
[w,s]=meshdom(w,s);
re=s.^2-w.^2+s+1; im=2*s.*w+w; den=re+j*im;
z=ones(den)./abs(den); mesh(z), grid
```

注意，上述程序中使用了矩阵元素对元素的运算命令。图形见图1-7。我们还可以改变命令中的参数从不同的角度来观察其网状表面图形。例如：

》mesh（z，[azimuth   elevate]）

上述命令允许你指定观察的方位与正视的角度。azimuth(方位)为正值时表示物体顺时针方向旋转；elevate（正视角度）为正值时表示从上方观察物体图形（90°为正上方）。缺省值是[−37.5 30]。

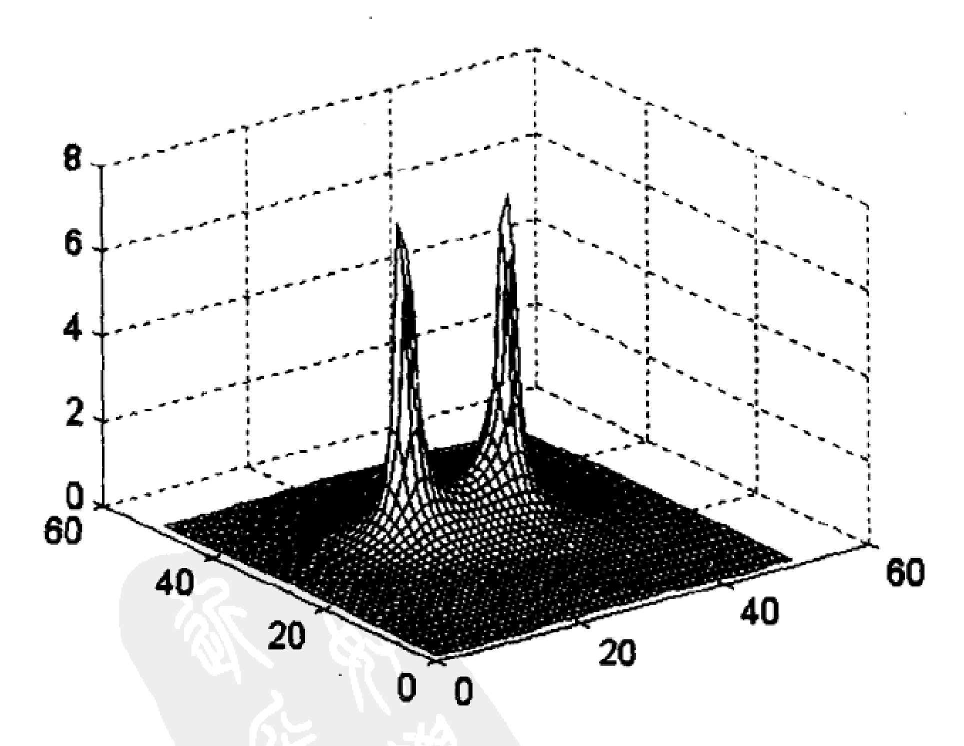


图1-7　频率响应的网状立体图

下面的例子采用直接构造一个区域矩阵（验证meshdom命令）的方法，并且从不同的角度观察这个物体图形。图形函数为

$z = \sin\sqrt{x^2 + y^2}$ ，MATLAB程序1-8为绘制三维立体图的程序清单。

```
% MATLAB PROGRAM 1-8
x=[-3: 0.1: 3];
y=x'; x2=ones(y)*x; y2=y*ones(x);
% 相当于meshdom的作用
r=sqrt(x2.^2+y2.^2);
z=sin(r); mesh(z), grid
```

得到的图形见图1-8。若再输入下面的命令

```
» mesh (z,[-30 +70])
```

可得到上面图形的旋转图形（从图形上方观看），见图1-9。

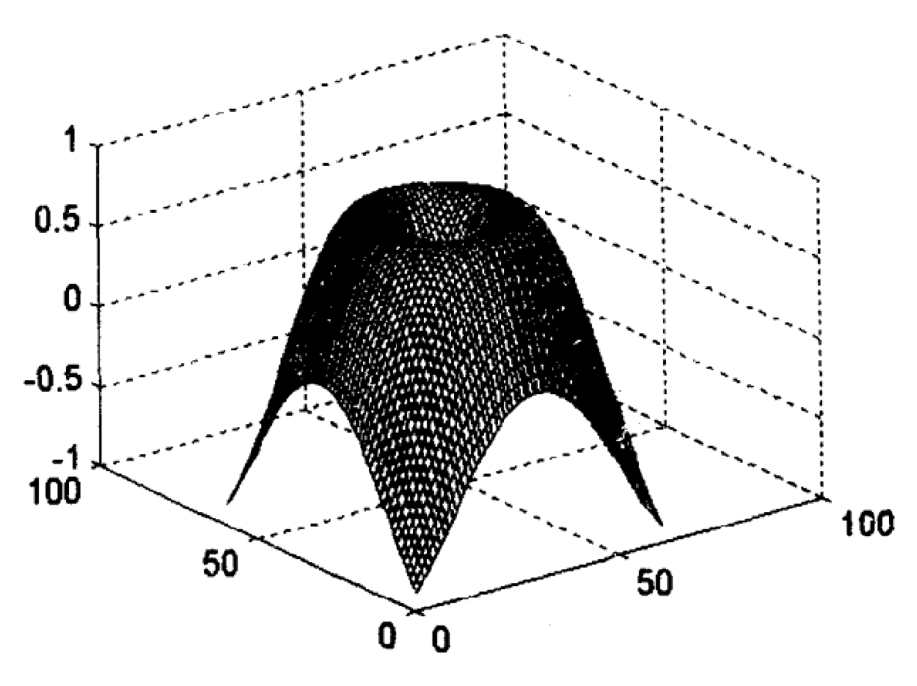


图1-8　三维立体图示例

除我们讨论的基本绘图命令外，MATLAB还有其它种类的绘图命令。如：bar（直方图）、stairs（梯形图）和contour(水平轮廓图) 等。在此就不一一介绍了，请见本章的附录及有关参考书，或者在MATLAB命令窗口下使用help命令，学习有关命令的用法。

本章简要地介绍了MATLAB语言及各种命令的使用方法。分

别介绍了MATLAB文件管理、数据结构、数学运算、多项式及MATLAB绘图的一些规则及使用方法。最后又给出了一些例题进一步说明各种命令的用法。希望读者通过本章的学习能够掌握MATLAB的基本用法。当然这需要一定量的上机练习才能达到。

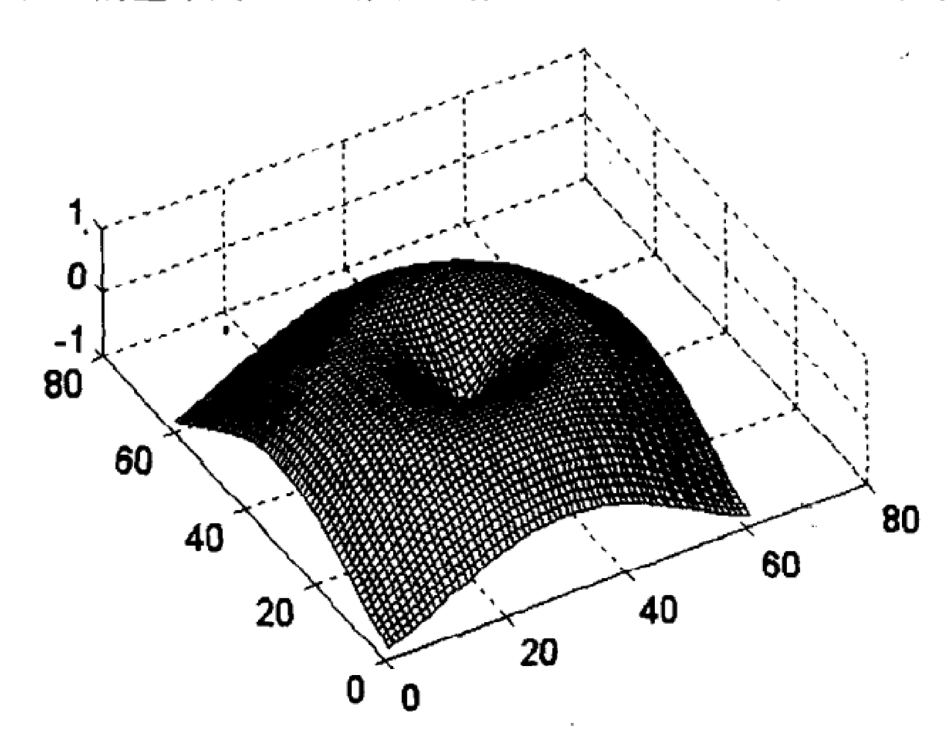


图1-9　改变方位后的立体图

习　题

1-1　用MATLAB命令完成矩阵的各种运算。已知矩阵

$$A=\begin{bmatrix} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ 31 & 32 & 33 & 34 \\ 41 & 42 & 43 & 44 \end{bmatrix}$$

求出下列的运算结果,并上机验证。

(1) **A**(:,1)　(2) **A**(2,:)　(3) **A**(:,2:3)

(4) **A**(2:3,2:3)　(5) **A**(:,1:2:3)　(6) **A**(2:3)

(7) **A**(:)　(8) **A**(:,:)　(9) ones(2,2)

(10) eye(2)　(11) [**A**,[ones(2,2); eye(2)]]

(12) diag(**A**)　(13) diag(**A**,1)　(14) diag(**A**,−1)

(15) diag(A,2)

1-2 用MATLAB命令完成下列矩阵函数运算：

(1) 输入如下矩阵A:

$$A=\begin{bmatrix}0 & \pi/3\\ \pi/6 & \pi/2\end{bmatrix}$$

(2) 求矩阵B_1,B_1中每一元素为对应矩阵A中每一元素的正弦函数。

(3) 求矩阵B_2,B_2中每一元素为对应矩阵A中每一元素的余弦函数。

(4) 求$B_1^2+B_2^2$。

(5) 求矩阵A的特征值与特征矢量；称特征矢量矩阵为M，而特征值矩阵为L。

(6) 求$M\sin(L)M^{-1}$。

(7) 使用funm命令求矩阵A的正弦函数。应该得到与(6)相同的结果。

(8) 求$\cos A$。

(9) 证明$\sin^2 A+\cos^2 A=I$

1-3 按题目要求用MATLAB命令完成下列矩阵运算。

(1) 使用rand命令产生5个2×2随机矩阵 A，B，C，D，E。

(2) 求下面的矩阵(不使用inv命令)

$$F=A^{-1}[B+C^{-1}(D^{-1}E)]$$

(3) 不使用inv命令，求矩阵A^{-1}的第1列（仅限使用1个命令）。

使用inv命令验证(2)与(3)的结果。

1-4 通过完成下列运算熟悉元素对元素的运算方法及kronecker乘法。每一部分先用手算，然后再用MATLAB命令上机验证。现有矩阵为

$$A=\begin{bmatrix}1\\1\\1\end{bmatrix}\qquad B=\begin{bmatrix}2 & 3 & 4\end{bmatrix}$$

（1）kron(A，B')

（2）kron(A，B)

（3）kron(A'，B)

（4）kron(A'，B')

（5）$A.*B'$

（6）$A.\backslash B'$

（7）$A'\backslash B$

1-5 绘制下列各种函数图形。

（1）绘制下列极坐标图形（区间$0\leqslant\theta\leqslant 2\pi$）：

1）$r=3(1-\cos\theta)$

2）$r=2(1+\cos\theta)$

3）$r=2(1+\sin\theta)$

4）$r=\cos 3\theta$

5）$r=e^{\theta/(4\pi)}$

（2）求函数z的三维图形。定义区间与z函数表达式如下：

$$-5\leqslant x\leqslant 5,\quad -5\leqslant y\leqslant 5$$

$$z=\frac{1}{(x+1)^2+(y+1)^2+1}-\frac{1.5}{(x-1)^2+(y-1)^2+1}$$

1-6 用MATLAB命令绘制下列数学函数的图形。

（1）绘制如下函数图形

$$y(t)=1-2e^{-t}\sin(t)\qquad (0\leqslant t\leqslant 8)$$

且在x轴写上“Time”标号，y轴上写“Amplitude”标号，图形的标题为“Decaying-oscillating Exponential”。

（2）绘制如下图形。

$$y(t)=5e^{-0.2t}\cos(0.9t-30^\circ)+0.8e^{-2t}\qquad (0\leqslant t\leqslant 30)$$

（3）在 $0\leqslant t\leqslant 10$ 区间内绘制下面图形。

$$y(t)=1.23\cos(2.83t+240^\circ)+0.625$$

$$x(t)=0.625$$

在同一个图中绘制上述函数，求出$y(t=0)$和$y(t=10)$的点。注意弧度与角度的区别。

（4）在 $0\leqslant t\leqslant 20$ 区间内，且在同一图中绘制下述函数图形。

$$y_1(t)=2.62e^{-0.25t}\cos(2.22t+174^\circ)+0.6$$

$$y_2(t)=2.62e^{-0.25t}+0.6$$

$$y_3(t)=0.6$$

在y值对应的−2到3的区域内，首先求y_1的最小值与最大值；然后再求次

大值与次小值。

（5）对应 $0 \leqslant t \leqslant 25$ 区域内，在同一图中绘制下例函数。

$$y_1(t)=1.25\mathrm{e}^{-t}$$

$$y_2(t)=2.02\mathrm{e}^{-0.3t}$$

$$y_3(t)=2.02\mathrm{e}^{-0.3t}\cos(0.554t-128^\circ)+1.25\mathrm{e}^{-t}$$

限制在如下区域内：

$$0.2 \leqslant y \leqslant 1 \qquad 0 \leqslant x \leqslant 16$$

求$y_3(t)$值：$y(t=0)$、y_{max}、y_{min}和$y(t=12)$。

1-7　完成下列各项要求，熟悉线性变换的作用。

在二维平面上一点可以用矢量表示（矢量$\boldsymbol{R}^2$空间的一个元素）。使用矩阵可以进行几何的线性变换（从一个二维空间到另外一个二维空间）。因此，$\boldsymbol{y}=\boldsymbol{Ax}$是$\boldsymbol{x}$经矩阵$\boldsymbol{A}$变换后平面上的另外一点或另外一个矢量。不同的矩阵具有不同的作用。它们可以扩充、压缩、映射、旋转或者进行其它矢量几何运算。

采用下列矩阵分别对在例题1-4所示的三角型进行变换，并观察其转换作用。在每一种情况下，首先绘制原来的三角型，然后在同一图中选取合适的标度绘制变换后的三角形。

(1) $\boldsymbol{A}_1=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$

(2) $\boldsymbol{A}_2=\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$

(3) $\boldsymbol{A}_3=\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$

(4) $\boldsymbol{A}_4=\begin{bmatrix}1 & 2\\ 1 & 0\end{bmatrix}$

(5) $\boldsymbol{A}_5=\begin{bmatrix}\cos(\pi/2) & -\sin(\pi/2)\\ \sin(\pi/2) & \cos(\pi/2)\end{bmatrix}$

注：选择适当的坐标刻度，以适应于两个图形同存于一个屏幕上的要求。

1-8　用MATLAB命令构造一个101个元素的矢量，它的元素交替为1和−1。使用plot、comb和stairs命令分别绘制该矢量的图形。

附录　MATLAB(Version 4.2c)常用函数命令一览表

字 符 名	说　　明
+	加
–	减
*	矩阵乘
.*	数列乘
^	矩阵乘方
.^	数列乘方
\	左乘
/	右乘
./	数列除
:	冒号
()	圆括号
[]	方括号
.	小数点
..	根目录
...	延续号
,	逗号
;	分号
%	评语号
!	惊叹号
=	赋值
==	等于
< >	关系算子
&	逻辑与
|	逻辑或
~	逻辑非
xor	逻辑异或

(续)

逻辑特征命令	说　　明
exist	检验变量或文件是否已经定义
any	测试矢量中是否有为真元素
all	测试矢量中所有元素是否为真
find	查找非零元素的下标
isnan	若参数为NaN，则结果为真
isinf	若参数为Inf,则结果为真
finite	若参数为有限元素则为真
isempty	若参数为空矩阵，则结果为真
issparse	若矩阵为稀疏，表示为真
isstr	若参数为字符串，则结果为真
isglobal	若参数为全局变量则为真
常　用　命　令	**说　　明**
help	启动联机帮助文件显示
what	列出当前目录下的有关文件
type	列出.M文件
lookfor	对help信息中的关键词查找
which	找出函数与文件所在的目录名
demo	运行MATLAB演示程序
path	设置或查询MATLAB的路径
管理变量工作空间的命令	**说　　明**
who	简要列出工作空间变量名
whos	详细列出工作空间变量名
load	从文件中读入变量
save	将工作空间中变量存盘
clear	删除内存中的变量与函数
pack	整理工作空间的内存
size	查询矩阵的维数

(续)

管理变量工作空间的命令	说　　明
disp	显示矩阵和文本
length	查询矢量的维数
文件与操作系统命令	**说　　明**
cd	改变当前的工作目录
dir	列出当前目录的内容
delete	删除文件
getenv	获得环境参数
unix	执行操作系统命令并返回结果
diary	将MATLAB运行的命令存盘
对命令窗口控制的常用命令	**说　　明**
cedit	设置命令行编辑与回调的参数
clc	清除命令窗口显示
home	将光标移动到左上角位置
format	设置输出格式
echo	显示文件中的MATLAB命令
more	控制命令窗口的输出页面
启动与退出MATLAB命令	**说　　明**
quit	退出MATLAB环境
startup	MATLAB自启动文件
matlabrc	启动主程序

矩阵分析函数	说　　明
cond	求矩阵的条件数
norm	求矩阵的范数
rcond	LINPACK倒数条件件估计
rank	求矩阵的秩
det	求矩阵的行列式

(续)

矩阵分析函数	说 . 明
trace	求矩阵的迹
null	右零空间
orth	正交空间
rref	矩阵的行列阶梯实现

线性方程函数	说 明
chol	Cholesky分解
lu	矩阵的三角（LU分解）
inv	矩阵求逆
qr	矩阵的正交三角化（QR）分解
qrdelete	QR分解中删除一列
qrinsert	QR分解中插入一列
nnls	非零最小二乘
pinv	伪逆矩阵
lscov	最小二乘方差

特征值函数	说 明
eig	求矩阵的特征值与特征矢量
poly	求矩阵的特征多项式
hess	求取Hessenberg标准型
qz	广义特征值问题求解（QZ算法）
rsf2csf	实块对角阵转换复块对角阵
cdf2rdf	复块对角矩阵到实块对角矩阵转换
schur	Schur分解
balance	改进特征值精度的均衡变换
svd	奇异值分解（SVD）

矩阵函数	说 明
expm	矩阵指数函数
expm1	expm 函数的.M文件实现

(续)

矩阵函数	说　　明
expm2	Taylor 级数法求矩阵指数
expm3	特征值特征矢量法求矩阵指数
logm	矩阵的对数
sqrtm	矩阵的平方根
funm	矩阵的任意函数
基本数学函数	**说　　明**
sin	正弦函数
sinh	双曲正弦函数
asin	反正弦函数
asinh	反双曲正弦函数
cos	余弦函数
cosh	双曲余弦函数
acos	反余弦函数
acosh	反双曲余弦函数
tan	正切函数
tanh	双曲正切函数
atan	反正切函数
atan2	四个象限内反正切函数
atanh	反双曲正切函数
sec	正割函数
sech	双曲正割函数
asec	反正割函数
asech	反双曲正割函数
csc	余割函数
csch	双曲余割函数
acsc	反余割函数
acsch	反双曲余割函数

(续)

基本数学函数	说　　明
cot	余切函数
coth	双曲余切函数
acot	反余切函数
acoth	反双曲余切函数
log	自然对数函数
log10	常用对数函数
sqrt	平方根函数
abs	绝对值函数
angle	相角函数
imag	求取虚部函数
real	求取实部函数
fix	对零方向取整数
floor	对−∞方向取整数
ceil	对+∞方向取整数
round	截取到最近的整数
rem	除法余数
sign	符号函数

基本矩阵与矩阵变换函数	说　　明
zeros	产生零矩阵
ones	产生元素全部为1的矩阵
eye	产生单位矩阵
rand	产生随机矩阵
randn	产生正态分布随机阵
linspace	构造线性分布的矢量
logspace	构造等对数分布的矢量
ans	返回最新结果

(续)

基本矩阵与矩阵变换函数	说　　明
eps	浮点相对误差限（2.2204×10^{-16}）
relmax	最大浮点数
realmin	最小浮点数
pi	圆周率π
NaN	不定式
flops	浮点运算计数器
nargin	函数实际输入变量个数
nargout	函数实际输出变量个数
clock	时钟
cputime	所用的CPU时间
date	日期
etime	所用时间的函数
diag	建立对角矩阵或获取对角矢量
flipud	按上下方向翻转矩阵元素
reshape	改变矩阵的行列数目
rot90	将矩阵元素旋转90°
tril	提取矩阵的下三角部分
triu	提取矩阵的上三角部分

数据分析与Fourier变换函数	说　　明
max	求矢量中最大元素
min	求矢量中最小元素
mean	求矢量各元素均值
median	求矢量各元素中间值
std	求矢量中各元素标准方差
sort	对矢量中各元素排序
sum	对矢量中各元素求和
prod	对矢量中各元素求积

(续)

数据分析与Fourier变换函数	说　　　明
cumsum	各元素累加和
cumprod	各元素累加积
trapz	梯形法求数值积分
diff	差分函数与近似微分
gradient	近似梯度计算
del2	五点式离散Laplace变换
corrcoef	相关函数系数
cov	协方差矩阵
subspace	子空间
filter	一维数字滤波
filter2	二维数字滤波
conv	求多项式乘法的卷积
conv2	二维卷积
deconv	因式分解与多项式除法
fft	离散Fourier变换
fft2	二维离散Fourier变换
ifft	离散Fourier逆变换
ifft2	二维离散Fourier逆变换
unwrap	除去每360°的跳跃
fftshift	取消谱中心零位
nextpow2	找出下一个2的指数

一般的图形函数	说　　　明
gcf	获得当前图形的窗口句柄
clf	清除当前图形窗口
close	关闭图形窗口
subplot	将图形窗口分成若干个区域
axes	坐标轴任意形式的设定

（续）

一般的图形函数	说　　明
gca	获得当前坐标轴的句柄
cla	清除当前坐标轴
axis	坐标轴标度设定
caxis	伪颜色坐标轴标度设定
hold	当前图形保护模式
两维图形函数	**说　　明**
plot	线性坐标图形绘制
loglog	全对数坐标图形绘制
semilogx	x轴半对数坐标图形绘制
semilogy	y轴半对数坐标图形绘制
fill	绘制充填的二维多边形
polar	极坐标图形绘制
bar	条形图曲线绘制
stem	离散序列柄状图形绘制
stairs	阶梯图形绘制
rose	极坐标（角度）直方图绘制
compass	绕行曲线绘制
fplot	给定函数绘图
title	给图形加标题
xlabel	给图形加x坐标说明
ylabel	给图形加y坐标说明
text	在图形上加文字说明
gtext	在鼠标指定的位置加文字说明
grid	给图形加网格线
三维图形函数	**说　　明**
plot3	绘制三维线或点型图形
fill3	绘制充填的三维多边形

(续)

三维图形函数	说　　明
comet3	绘制三维彗星状的轨迹
contour	等高线图形绘制
contour3	三维等高线绘制
pcolor	伪颜色绘图
mesh	三维网格图形
meshc	具有等高线的网格图形
meshz	带有零平面的三维网格图形
surf	三维表面图形
surfc	具有等高线的三维表面图形
view	三维图形视口指定
hidden	网格图隐含线设置开关
shading	阴影模型
字符串函数命令	**说　　明**
strings	关于MATLAB字符串的帮助信息
setstr	将数值转换为字符串
blanks	设置一个由空格组成的字符串
str2mat	字符串转换为矩阵
eval	执行MATLAB语句构成的字符串
strcmp	字符串比较
findstr	由一个字符串中查找
upper	将一个字符串内容转换为大写
lower	将一个字符串内容转换为小写
isletter	若字符串为字母组成则为真
num2str	将数值转换为字符串
int2str	整数转换为字符串
str2num	字符串转换为实型数据
hex2num	十六进制到IEEE浮点数的转换

第2章　MATLAB语言的编程方法

MATLAB语言的编程结构与其它语言编程结构相似，由一些基本结构组成，如循环与条件分支结构等。这些结构的语言内容及用法与其它高级语言(如C、BASIC和FORTRAN)非常相似。由这些新命令组合的程序，可以得到功能更强的MATLAB函数。这些新的函数又可以加入到MATLAB命令中。本章将着重讨论这些特点。

2.1　关系与逻辑运算符

关系运算允许常量(或矩阵中的元素与元素)之间的比较，其比较结果为0或者为1。如果比较结果为真，则答案为1，否则为0。常用的关系运算符如下：

< 小于	<= 小于或等于	== 等于
> 大于	>= 大于或等于	~= 不等于

对于复数运算，“==”与“~=”运算，既比较实部，又比较虚部。而其它运算仅比较实部。也可以用一常量与矩阵比较，在这种情况下，该元素与矩阵的每一个元素进行比较。结果为一个由0和1组成的矩阵。下面让我们看一些实例。请注意，有时为了表达清楚起见，加入一些空格，然而需要注意它们的正确用法。例如：x >= y，这样空格是正确的。但是x> =y是错误的，即在两个符号之间不能加入空格。请看下面的例子：

```
» a=[1+j pi 2 3.14 1-i]; b=[1-i 3.14 2 3.14 1+i];
» c=a>b
    c=
        0 1 0 0 0
» c=a==b
```

```
c=
        0 0 1 1 0
》 c=a>=b
   c=
        1 1 1 1 1
》 c=a~=b
   c=
        1 1 0 0 1
```

MATLAB语言中有三个逻辑运算符用于逻辑关系运算，它们是：

&为逻辑与　　　　|为逻辑或　　　　~为逻辑非

X&Y的结果为(0,1)矩阵，该矩阵与矩阵***X***和***Y***具有同样的行列数。若***X***与***Y***对应的相同位置的元素都不为零，则比较后得到的那个矩阵中相应位置元素为1，否则为0；***X|Y***运算结果为1时，表示该位置的***X***的元素或***Y***的元素不为0，否则为0(即两个元素均为0)。~***X***运算结果为1时，表明该位置元素为0，否则运算结果为0，例如：

```
》 X=[0, 2; pi, 0]; Y=[0, 0; j, 0];
》 X&Y
  ans=
        0   0
        1   0
》 X|Y
  ans=
        0   1
        1   0
```

它们的运算顺序为算术运算、关系比较，然后为逻辑运算。例如(***X&Y+X>Y***)，***X***与***Y***相加，相加结果与***Y***比较，比较的结果与***X***做与运算。

```
》 Z=X&Y+X>Y
```

```
Z=
        0  1
        1  0
》Z=X>Y+X&Y
 Z=
        0  0
        0  0
```

在MATLAB语言中有两个逻辑命令，它们也可以产生(0,1)矩阵。这两个命令为any和all。

```
》any(X)
```

对应矢量x不为零时结果为1，否则为0。对于矩阵，该运算是逐列进行的，即该列矢量不为零，则为1，否则为0。例如:下列为两个矩阵的比较，如果它们不相等结果为1。

```
》any(any(A~=B))
```

all命令的运算功能为，如果矢量x的所有元素不为0，其结果为1，否则为0。对于矩阵运算是逐列进行的。

```
》all(X)
```

例如，如果下列两矩阵相等，则下列结果为1。

```
》all(all(A==B))
```

如果一个矩阵等于本身的转置矩阵，则该矩阵为对称矩阵。命令all可以用来判定一个矩阵是否为一个对称矩阵，如果矩阵是对称的，其结果为1。例如：

```
》all(all(A==A'))
 ans=
        1
```

最后，介绍一个很常用的命令find，它可以找出一个矢量或一个矩阵中非零元素的位置。其命令格式为

```
》find(X)
```

该命令可得到矢量x中非零元素的序号。如果参数是一个矩阵，MATLAB用X(:)把它转换成为一个长列。例如，有一个用若

干个0开头的矢量，假设要去掉前面的0，可采用如下命令：

```
》ind=find(p~=0); p_=(ind(1): length(p));
```

上面第一条命令可找出非0元素的位置；第二条命令可重新定义这个矢量。例如：

```
》p=[0 0 1 3 2]; ind=find(p~=0);
》p_=p(ind(1):length(p)); p_
  p_=
       1  3  2
```

下列命令可以得到一个矢量或一个矩阵中非0元素的个数。

```
》nonzer=length(find(X))
```

MATLAB语言还有一些检测非数值(NaN)、空矩阵、无穷大值和字符串的命令。这些命令是isnan、isempty、isstr和finite。

当它们找到NaN，空矩阵和字符串时，isnan、isempty和isstr命令返回值为1；当矢量中含有无穷大值时，finite命令返回值为0。

从以下的命令可以找出一个矢量中最大元素的位置。这个功能在确定阶跃响应的峰值时间时很有用处。假设矢量x已经被定义，则

```
》peaktime=find(X==max(X))
```

式中，peaktime为峰值时间。

2.2 循环与条件结构

MATLAB语言有三个命令用于循环、条件循环与条件叙述程序设计结构中。它们是for、while和if else命令。循环可以嵌套，陈述句可以跨行，为清楚起见，也可以使用空格。所有的命令必须用end作为结束命令。在循环嵌套中，上述每一个命令后面都需要用end作为结束语句。

for ：for命令用于执行一系列重复的陈述命令。它的格式内容显示如下：

```
》for  变量=表达式，陈述句...陈述句，end
```

表达式是一矢量或是一个矩阵，或是由MATLAB命令产生的一个矢量或一个矩阵。陈述语句执行一次对应这个行矢量的一个元素或矩阵的一列。循环结构会使程序变慢。因此要尽可能采用矢量与矩阵运算的优点而代之。如sort、find、any、all命令和各种矩阵地址运算。

【例2-1】采用不同的方法计算从1到10000数字的对数。然后比较不同的方法的计算机运算时间。时钟clock与etime命令用于测定采用不同方法时计算机运算的时间。MATLAB程序2-1为采用不同的方法计算从1到10000数字的对数的程序清单。

```
% MATLAB PROGRAM 2-1
t1=clock; for i=1:10000, a(i)=log(i); …
end; e1=etime(clock,t1)
t1=clock; ind=[1:10000]; for i=ind, a(i)=log(i); …
end; e2=etime(clock,t1)
t1=clock; a=zeros(1:10000); ind=[1:10000]; …
for i=ind,a(i)=log(i); end; e3=etime(clock,t1);
t1=clock; ind=[1:10000]; a=log(ind); …
e4=etime(clock,t1);
t1=clock; ind=[1:10000]; a=zeros(1,10000); …
a=log(ind); e5=etime(clock,t1)
```

上面第一种方法直接采用for循环指令。在第二种方法中，将循环序号预先定义为一个行矢量，然后按矢量列执行循环运算。在第三种方法中，存储运算结果的矢量a是预先定位的。可以看出，运算的时间急剧减少。执行时间减少的主要原因在于，在前面的方法中，每一循环MATLAB都需要重新确定矢量的大小，而在第三种方法中，采用预先定位方式，因此在每一循环中都略去了这一步运算。在第四种方法中，删去了for循环命令，而使用MATLAB语言的矩阵计算功能，可以看出，运算的时间又一次明显地减少。在最后一种方法中，利用了MATLAB语言的矩阵运算能力，又使用了预先定位的方法。可以看出，其计算时间稍微有

所增加。这时因为没有使用循环结构，根本不需要重新确定矢量的大小，因此在这种情况下，预先定位是不必要的。在MATLAB用户指南中指出，输出变量的预先定位会提高存储器的利用率。因此，对大矩阵进行数据运算时，建议使用输出变量的预先定位方法。特别是当计算机的存储容量有限时，更应该如此。

【例2-2】 假设要交换两个矩阵的元素。给定矩阵***A***和***B***，希望矩阵***A***的每隔一行等于矩阵***B***每隔一行的最后七个元素。其程序如下：

```
》a=[1:2:13]; am=ones(7); for i=1:7,am(i,:)=a; ...
end; am
    am=
         1   3   5   7   9   11   13
         1   3   5   7   9   11   13
         1   3   5   7   9   11   13
         1   3   5   7   9   11   13
         1   3   5   7   9   11   13
         1   3   5   7   9   11   13
         1   3   5   7   9   11   13
》b=0:2:18; bm=ones(10); for i=1:10, bm(i,:)=b; ...
end; bm
    bm=
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
         0   2   4   6   8   10   12   14   16   18
```

0 2 4 6 8 10 12 14 16 18

下面的程序是采用for循环结构的方法。

```
» a2=am; for i=1:2:7, for j=1:7, ...
a2(i,j)=bm(i+1,j+3); end; end; a2
   a2=
        6   8   10   12   14   16   18
        1   3   5    7    9    11   13
        6   8   10   12   14   16   18
        1   3   5    7    9    11   13
        6   8   10   12   14   16   18
        1   3   5    7    9    11   13
        6   8   10   12   14   16   18
```

采用矩阵地址的方法，可以得到同样的结果。

```
» a2=am; a2([1:2:7], [1: 7])=bm([2:2:8], [4:10])
```

while：while命令可构成条件循环结构，在循环过程中，只要条件满足就执行其陈述语句命令。它的命令格式为

```
» while  表达式, 陈述句, ..., 陈述句, end;
```

其中，表达式的格式为 x 运算符 y，x与y为常量或由运算公式产生的常量。运算符通常为关系运算符。

【例2-3】使用while命令寻找一个随机的而且是稳定的矩阵(即它的特征值实部为负值)的例子。其MATLAB程序如下：

```
% MATLAB PROGRAM 2-2
rand('normal'); a=rand(2);
while max(real(eig(a)))>=0, a=rand(2); end; eig(a)
   ans=
      -0.2185+1.0137i
      -0.2185-1.0137i
```

【例2-4】使用while命令计算并显示著名的费班纳赛

(Fibonacci)数。费班纳赛数定义如下:

$$f(n+2)=f(n+1)+f(n) \qquad f(0)=1, f(1)=1$$

MATLAB命令如下:

```
》f[1 1]; i=1; while f(i+1)+f(i)<100;
f(i+2)=f(i+1)+f(i); i=i+1; end; f
  f=
    1 1 2 3 5 8 13 21 34 55 89
```

if、else、elseif：if、else、elseif等命令也用于条件执行语句中，它们的格式内容为

```
》if 表达式1，陈述句，...，陈述句，...
     elseif 表达式2，陈述句，...，陈述句，...，
     else   陈述句，...，
  end
```

如果表达式1为真，则执行第一套陈述句；如果表达式2为真，则执行第二套陈述句；否则执行else后面的陈述句。此处表达式具有与while陈述句中的相同结构，即通常为一关系运算符。elseif与else的部分是可选择的。注意以上三个命令都是可以嵌套的。但不要忘记，在程序的末尾使用end命令作为结束语句。下面是两个应用的例子。

第一个例子是去掉多项式前面的零项。在前面使用find命令时已采用过这个例子。其程序为

```
》p=[0 0 0 1 3 0 2 0 0 9];
》for i=1:length(p),if p(1)==0,p=p(2:length(p));
  end;
  end; p
  p=
    1  3  0  2  0  0  9
```

在下面另外一个例子中，做与上例相反的工作。首先，比较两个矢量的元素数量；如果它们元素数量不同，在元素数量小的

矢量中插入0元素，使得两个矢量的元素数量相等：如果原来两个矢量的元素数量是相等的，则保持原矢量不变。

```
》p=[1 2 3]; q=[1 2 3 4 5 6];
》lp=length(p); lq=length(q);
》if lp==lq, break
        elseif lp>lq
        q=[zeros(1,lp-lg)q]
        else p=[zeros(1,lq-lp)p]
        end
        p=0  0  0  1  2  3
```

注：break命令可以中止其它命令的执行。

2.3 M文件：程序文件与函数文件

在一般方式中，MATLAB命令是立即执行方式，即输入命令后MATLAB立即响应。MATLAB也可以执行存储在文件中的程序。只要该文件满足两个条件：其一，程序驻存的文件(ASCII码文件)必须在MATLAB当前目录下；其二，该文件必须有.m的扩展名。例如，上述给矢量前面加零的命令程序可以用"zeropad.m"文件名存储。然后再执行该程序，这就是为什么称MATLAB程序为M文件的原因。通常使用任何可以生成ASCII码文件的文本编辑器都可以生成M文件。几乎任何文本编辑器都具备这个特点。一定要确定文件是ASCII码文件形式，否则MATLAB将不能确认。MATLAB中有两种形式的M文件：程序文件与函数文件。

2.3.1 程序文件

当完成一个功能需要许多MATLAB命令时，可以把这些命令存储在一个程序文件中。然后在MATLAB提示符下，输入程序文件名，回车即可执行这些命令。文件中的命令是对整体工作空间的数据进行运算的。例如，如果x已在工作空间，在运行文件中重新定义x，x值将按照新定义值改变，在文件中所用到的变量需

要在工作空间中事先定义，或者在文件中重新定义。

例如，将下列命令存入以“padzer.m”为文件名的文件中。

```
lp=length(p); lq=length(q);
if lp==lq, break
  elseif lp>lq
    q=[zeros(1,lp-lq) q]
 else p=[zeros(1,lq-lp) p]
end
```

在使用该文件前，应先定义p与q；然后执行该命令。

```
》p=[1 2]; q=[1 2 3 4 5];
》padzer
```

2.3.2 函数

函数是MATLAB扩展功能的M文件。许多MATLAB命令与全部工具箱命令都是用M文件格式写成的。程序文件与函数文件的主要区别在于，函数允许通过数值进行参数传递，而且函数使用局部变量而不是对整个工作空间中变量的操作。另外一个不同点是，函数文件的第一行必须包括“function”这个字。下面是一个典型的函数文件：

```
function[out1,out2,...]=filename(in1,in2,...)
% optional comment lines for documentation
MATLAB commands
```

函数可以有多输入和多输出形式。与其它MATLAB函数相似，输入参数放在小括号内，而输出参数放在中括号内(如果只有一个输出，则没有必要使用括号)。在第一行之后，可以写入一些有关函数的说明，当使用帮助(help)命令时，则可以显示这部分说明。在函数文件中，定义的所有变量都是内部变量，函数执行完毕后全部消失。通过工作空间传递的变量仅仅分配给输出参数。在函数内部的变量不会影响整个工作空间。函数文件可以像其它MATLAB命令一样被使用。

下面是一个简单函数“cltf”的例子。假设传递函数$G(s)$与

$K(s)$在单位反馈结构中是串联的。由该函数可求出开环传递函数$G(s)K(s)$与闭环传递函数$T(s)$。

```
function[ngk, dgk, nt, dt]=cltf(ng, dg, nk, dk)
ngk=conv(ng, nk); dgk=conv(dg, dk);
dimngk=length(ngk); dimdgk=length(dgk);
nc=[zeros(1, dimdgk -dimngk), ngk];
dt=nc+dgk
nt=ngk
```

【例2-5】 使用函数"cltf"求取单位反馈系统的闭环传递函数。

$$G(s)=\frac{1}{s^2} \qquad K(s)=\frac{s+5}{s+10}$$

求取系统闭环传递函数的MATLAB命令如下：

```
》ng=1;dg=[1, 0, 0, ];nk=[1, 5];dk=[1, 10]
》[ngk, dgk, nt, gt]=cltf(ng, dg, nk, dk)
  ngk=
     1   5
  dgk=
     1   10   0   0
  nt=
     1   5
  dt=
     1   10   1   5
```

2.3.3 建立M文件

在Windows的MATLAB环境下，可以使用笔记本(Notebook)编辑器建立M文件。在编辑环境中输入MATLAB命令，然后存盘。在这个环境中，可以在MATLAB与Notebook之间转换修改你的程序，直到得到一个满意的程序为止。在DOS版本的MATLAB环境下可以使用编辑器(edit)编写M文件。如果使用edit命令，在matlab.bat文件中必须含有一行执行文本编辑器的设定命令。在

edit命令前使用“!”符号可以使你在MATLAB环境中进入DOS的编辑环境。当从编辑中退出时，又自动地进入MATLAB环境。

有时当你不知道如何去解决一个问题时，大多数人不得不使用trial和error命令，或者使用“poking around”命令，直到他们得到一个满意的程序。在这种情况下，采用“diary on”命令能使程序修改更方便快捷。“diary”命令将你的输入与输出存储在一个文件中；然后可以使用文本编辑器去掉其中的错误，而得到一个完满的程序。

2.4 字符串宏命令

宏是MATLAB语言用在常用命令部分的缩写。它可以被存储用于将来建立M文件的一部分。宏命令采用字符串，并使用eval命令去执行宏命令。下例是采用宏命令计算阶乘的例子。

》fct= 'prod(1:n)';

求10的阶乘10!

》n=10；eval(fct)

ans=

362800

2.5 常用编程命令

MATLAB有几个命令能够使得程序具有交互性，使用这些命令有助于调试程序。下面简单讨论这些命令的功能。

》pause

停止M文件的执行直至有键按下。该命令常用于绘图命令之后；pause(n)命令将使程序暂停n(s)。pause(-2)将取消后续的停止命令。

》echo　或echo on　或echo off

当执行M文件时，显示程序的内容。echo是转换命令，因此输入一次echo将改变echo的状态。

》keyboard

该命令将停止执行M文件，把控制权交予键盘，可以观察与改变所有你希望的变量。输入return并按下return(或回车)键，将允许M文件继续执行。该命令对于调试程序是很有用的。

```
》x=input( 'prompt')
```

input命令把输入的字符串作为提示符，等待使用者输入一个响应，然后把它赋值到x。正是这个input命令使得你能够建立一个具有交互功能的程序。下例是一个具有交互功能的阶乘宏命令程序。

```
》f1= 'prod(1:n)';
》f2= 'n=input( "enter number:__"); z=eval(f1);
  disp("n!is"), disp(z)';
》eval(f2)
  enter number:__5
  n! is
      120
```

由上述程序可以得到下述几点。交互性宏f2可以调用另外一个宏f1。当你使用字符串含有撇号(')时，必须使用两个撇号(不要使用引号)。例如，宏命令f2是用撇号包括的；在input与disp命令中的字符串也是用撇号包括的；对应字符串的撇号都是双撇号。

```
》exit( 'item')
```

该命令的功能是检查工作空间的变量或MATLAB路径中的文件是否存在。结果是1时，表示该变量存在于工作空间；结果是2时，表示文件存在在MATLAB路径中；否则结果为0。

2.6 编程举例

【例2-6】 编写一个简单的M文件，计算二阶系统的阶跃响应。ζ值从0.1至1，且绘制出一簇阶跃响应图。

可以使用下列MATLAB程序计算并绘制二阶系统的阶跃响应曲线：

```
% MATLAB PROGRAM 2-3
n=1; y=zeros(200,1); i=1;
 for del=0.1:0.1:1,
      d=[1,2*del,1];
      t=[0:0.1:19.9]';
      y(:,i)=step(n,d,t);
      i=i+1;
end;
mesh(fliplr(y),[-120 30])
```
绘制的阶跃响应图见图2-1。

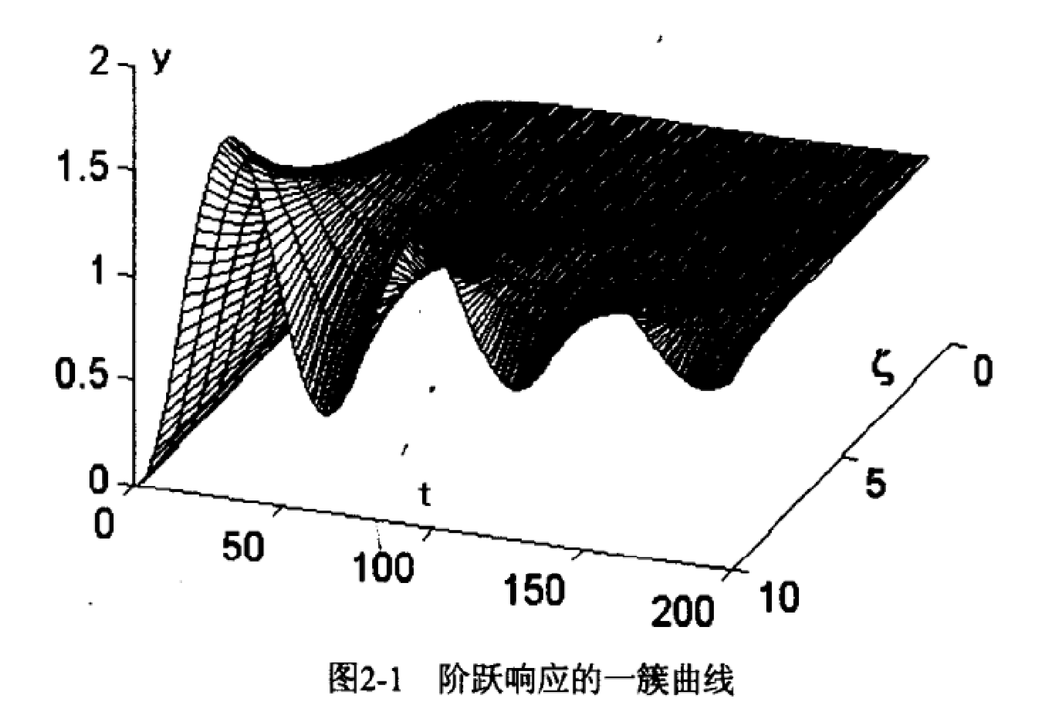


图2-1　阶跃响应的一簇曲线

【例2-7】 求一个系统所对应的三个不同参数的阶跃响应。即：

$$G(s)=\frac{b}{s^2+as+b}$$

式中，a={1,2,4}，b={1.25,2,29}。采用MATLAB程序2-4可求出对应的三个参数的阶跃响应。

```
% MATLAB PROGRAM 2-4
```

```
a=[1 2 4];  b=[1.25 2 29];
y=zeros(100,3);  t=linspace(0,10,100)';
for j=1:3
num=b(j),den=[1 a(j) b(j)],
y(:,j)=step(num,den,t);
end
```

【例2-8】编制一个函数程序计算阶跃响应的特征参数：(1)超调量σ %；(2)上升时间T_r；(3)峰值时间T_p；(4)过渡过程时间T_s（2%）。MATLAB程序2-5为计算阶跃响应的特征参数程序清单。

```
% MATLAB PROGRAM 2-5
funcation [pos,tr,ts2,tp] = stepchar(t,y)
% Finding Pos and Tp
[mp,ind]=max(y); dimt=length(t); yss=y(dimt);
pos=100*(mp-yss)/yss; tp=t(ind);
% Finding rise time Tr
i=1; j=1; k=1; q=1;
while y(i)<0.1;
        i=i+i;
end;
t1 = t(i);
while y(j)<0.9;
        j=j+1;
end;
t2 = t(j); tr=t2-t1;
% Finding settling time (two percent) Ts
i=dimt+1; n=0;
while n==0,
       i=i-1;
              if i==1,
```

```
                n=1;
                elseif y(i) >=1.02,
                n=1;
                end;
end;
t1=t(i);  i=dimt+1; n=0;
while n==0,
      i==i-1;
          if y(i) <= 0.98,
                n=1;
          end;
t2=t(i);
       if t1>t2;
             ts2=t1;
       else
             ts2=t2;
       end
end
```

程序中，Pos为超调量$\sigma\%$。

【例2-9】 使用for命令与if elseif结构编制上述程序。MATLAB程序如下：

```
% MATLAB PROGRAM 2-6
funcation [pos,tr,ts,tp]=stpchar2(t,y)
dimt=length(t); [mp,ind]=max(y); yss=y(dimt);
pos=100*(mp-yss)/yss; tp=t(ind);
for i=1: dimt,
       if y(i) > 1.02*yss,
              ts=t(i);
       elseif y(i) < 0.98*yss
              ts=t(i);
```

```
end
end
for i=1: dimt
        if y(i) < 0.1*yss
                t1=t(i);
        elseif y(i)==mp
                break;
        end
end
for i=1:dimt;
        if y(i) < 0.9*yss,
                t2=t(i);
        elseif y(i)==mp,
                break
        end;
end
tr = t2-t1;
```

习　　题

2-1　使用for、sum和prod命令，完成下列各项要求。

(1)利用prod命令生成与列写的N的阶乘($N!$)，并且对应N=1～10进行验证。

(2)使用for命令代替prod命令，完成（1）项中的功能。

(3)使用sum命令求与列写N个数的连续和。对应N=1～10进行验证。

(4)使用for命令代替sum命令，完成（3）项中的功能。

(5)编写一个程序，生成与列写第一个整数平方和，并且对应N=1～10进行验证。

2-2 结合本章范例，编写一函数程序，完成同时计算几个阶跃响应特征参数的功能。注：在本例中，阶跃响应数据是一列矢量；在此同时馈送几个响应，因此响应数据应是一个矩阵(每一列对应不同的阶跃响应数据)。

2-3 完成下列程序设计，比较各个系统的阶跃响应。

(1)求出下列传递函数的极点。利用习题2-2中的程序计算下列习题的阶跃响应特征参数($\sigma\%$、T_r、T_s、T_p)。

$$T_1 = \frac{2}{s^2 + 2s + 2}\ ,\ T_2 = \frac{4s + 2}{s^2 + 2s + 2}\ ,\ T_3 = \frac{1}{2s^3 + 3s^2 + 3s + 1}$$

(2)编制一个程序，求系统的频率响应(ω、mag、phase)，并求得输出频率特性(M_r、Bw)。

(3)使用上述程序，制表列写上述数据。

(4)比较加入零点(如T_2)与加入极点(如T_3)时的时域与频域的结果。

2-4 编制一个程序，该程序取传递函数的分子与分母作为程序的输入参数，然后求它的极点、独立的复数主导极点，并且计算它的阶跃响应特征参数。注：用该程序计算阶跃响应特征参数时，由于是利用主导极点与计算公式，因此与上述数值方法的结果有所区别。



第3章 经典控制系统分析的常用命令及SIMULINK 仿真软件简介

本章将介绍一些在经典控制系统分析中常用命令及SIMULINK 仿真软件。这些控制系统分析的常用命令被置于控制系统工具箱中。在自动控制系统分析中，主要讨论系统的脉冲响应、阶跃响应、一般输入响应、频率响应及由传递函数表示的系统根轨迹。传递函数 $G(s)$是由其分子的多项式与分母的多项式分别定义而确立的。 MATLAB 将这些多项式在命令中解释为传递函数。 SIMULINK（以前称为 SIMULAB ）是 MATLAB 程序的扩展。 SIMULINK 与 MATLAB 主要不同之处在于它是基于Windows 环境下的图形程序，为用户进行系统分析提供了极大的方便。由于它的工作方式特点，建议你要通过上机练习来学习这部分内容。

3.1 时间域命令

许多控制系统命令在没有引用左面变量（即输出变量）情况下，会自动绘制图形。基于极点与零点的位置，自动选取算法会找到最佳的时间或频率点。然而自动绘图的结果不会生成数据。这种命令适用于初始的分析与设计。对于深入问题的分析，应该使用带有输出变量形式的命令。

单输入单输出 SISO 系统 $G(s)$=num(s)/den(s)的阶跃响应 $y(t)$可以由 step 命令得到。命令格式如下：

```
》 y=step(num， den， t)
```

注意，时间 t 轴是事先定义的矢量。阶跃响应矢量与矢量 t 有相同的维数。对于单输入多输出(SIMO)系统，输出结果将是一个矩阵，该矩阵应有与输出数量相同数量的列。对于这种情况，

step 将有其它命令格式（请见第 5 章状态空间设计方法一章）。

例如，计算并绘制下面传递函数的阶跃响应(t=0 至 t=10)。

$$G(s)=\frac{10}{s^2+2s+10}$$

输入

》 num=10； den= [1， 2， 10]；

》 t= [0:0.1:10]'； y=step(num， den， t)； plot(t,y)

上列程序会生成 2 个列矢量、 101 个元素。一个对应时间轴，另外一个对应阶跃响应。

脉冲响应是通过使用 impulse 命令获得的。它的格式与 step 命令相似，即

》 y = impulse （ num， den， t ）

对应一般输入的响应也可以得到。命令为 lsim，其命令格式为

》 y = lsim （ num， den， u， t)

输入信号为矢量 $\boldsymbol{u}$。输入信号 $\boldsymbol{u}$ 的行数决定了计算的输出点数。对于单输入系统， $\boldsymbol{u}$ 是一个列矢量。对于多输入系统， $\boldsymbol{u}$ 的列数等于输入变量数。例如：计算斜坡响应， $\boldsymbol{t}$ 为输入矢量。

》 ramp = t； y = lsim （ num， den， ramp， t ）

我们也可以使用 rand 函数得到随机噪声响应。注意到 rand(m， n)生成一 $m\times n$ 的矩阵，其矩阵元素为在 0 ～ 1 之间的随机数。从下面给出的命令，可得到该系统 10s 的噪声响应。

》 noise = rand （ 101， 1 ）；

》 y =lsim （ num， den， noise， t ）；

3.2 频率域命令

使用 bode 、 nyquist 与 nichols 命令可以得到系统的频率响应。如果命令中没有使用输出变量，这些命令可以自动地生成响应图形。bode 命令的各种格式如下：

》 bode(num， den)

》 [mag， phase， w] =bode(num， den)

》 [mag， phase] =bode(num， den， w)

命令中 w 表示频率ω。上述第一个命令在同一屏幕中的上下两部分分别生成伯德幅值图(以 dB 为单位)与伯德相平面图（以 rad 为单位）。在另外的格式中，返回的幅值与相角值为列矢量。此时幅值不是以 dB 为单位的。第二种形式的命令自动生成一行矢量的频率点。在第三种形式中，由于用在定义的频率范围内，如果比较各种传递函数的频率响应，第三种方式显得更方便一些。对应于其它格式使用信息，请使用在线(on-line)帮助信息(help-bode)。

下述命令为绘图的其它命令

》 subplot(211)， semilogx （w， 20*log10(mag)），

》 subplot(212)， semilogx （w， phase）

上面的第一个命令把屏幕分成两个部分，并把幅值图放置在屏幕的上半部。第二个命令中使用 semilogx 命令生成一个半对数图（横轴是以 10 为底的对数值坐标轴，而竖轴是以 dB 为单位表示的幅值）。第二行命令将系统相频图放置在屏幕的下半部分。如果你想以 Hz 为单位，可用 w/2*pi 来代替 w 。如果你想指定频率范围，可以使用 logspace 命令：

》 w = logspace （m， n， npts）

该命令生成一个以 10 为底的对数矢量（ $10^m \sim 10^n$ ），点数(npts)是可选的。例如，下述命令生成 0.01 ～ 1000rad/s 的点：

》 w = logspace （-2， 3）；

nyqnist 与 nichols 命令有如下格式：

》 [re， im] =nyquist （num， den， w）

》 [mag， phase] = nichols （num， den， w）；

》 magdb=20*log10(mag)；

myquist 命令可计算 $G(j\omega)$的实部与虚部。在复平面上绘制虚部与实部的轨迹，亦可得到其奈魁斯特图形。 nichols 命令可计算幅值与相角值（以 rad 为单位）。如果你已经执行了 bode 命

令，可以通过绘制幅值与相角值直接得到相同的结果。使用 ngrid 命令可以在 Nichols 图上加画格线，即在提示符下输入 ngrid 。

使用 margin 命令可以求得相对稳定性参数（增益裕量与相角裕量）。它的命令格式为

```
》[ gm， pm， wpc， wgc ]= margin （ mag， phase， w ）
》 margin （ mag， phase， w ）
```

命令的输入参数为幅值（不是以 dB 为单位）、相角与频率矢量。它们是由 bode 或 nichols 命令得到的。命令的输出参数是增益裕量（不是以 dB 为单位的）、相角裕量（以角度为单位）和它们所对应的频率。第二个命令格式中没有左参数，它可以生成带有裕量标记的（垂直线）伯德图。如果在轴上有多个穿越频率，图中则标出稳定裕量最坏的那个标记。第一种命令格式就没有绘出最坏的裕量。请注意，用 margin 命令有时计算出的结果是不准确的。

3.3 根轨迹法命令

使用 rlocus 命令可以得到连续的单输入单输出系统的根轨迹。该命令有两种基本形式。

》 rlocus （ num， den ）或 rlocus （ num， den， k ）

在这些命令中，根轨迹图是自动生成的。如果这第三个参数（矢量 ***k*** ）是指定的，命令将按照给定的参数绘制根轨迹图，否则增益是自动确定的。

下面的命令可求得系统的闭环极点。可以通过使用你所选择的一个符号，绘制闭环极点的实部与虚部，得到一个系统的根轨迹图。

```
》 clpoles = rlocus （ num， den ）
 （或 clpoles = rlocus （ num， den， k)）
》 plot （ real(clpoles)， imag(clpoles),э*э ）
```

axis 命令可以定义绘制图形轴线的区域。定常阻尼系数ζ（从 0 至 1，间隔增量为 0.1 ）与自然频率的轨迹线可以通过使用 sgrid

命令绘制在同一个轨迹图上。

》 sgrid 或 sgrid(zeta， wn)

第二种形式允许你指定阻尼系数与自然频率的范围。下列命令为绘制系统 $G(s)$的根轨迹命令。绘制的区域为靠近虚轴的上半平面，且在平面上同时绘制阻尼比线（ζ从 0.5 至 0.7）与自然频率线（0.5rad/s）：

》 ng=1， dg= [1 3 2 0]； axis([-1 1 0 3])；

》 rlocus(ng, dg)

》 sgrid([0.5： 0.1： 0.7] ,0.5)

在系统的分析过程中，常常希望确定根轨迹上某一点的增益值。 rlocfind 命令就可以完成该项工作。第一步要得到系统的根轨迹，然后执行下面的命令：

》 [k， poles] =rlocfind （num， den）

执行命令后，将在图形屏幕上生成一个十字光标。使用鼠标器，移动这个十字光标到所希望的位置，然后敲左键，将得到该极点的位置及它所对应的增益 K 值。如果所选择的点接近于根轨迹上某点，则该点对应的增益值及极点位置将作为命令的输出参数。

该命令也可以在没有绘制根轨迹图之前执行。此时，使用命令的格式如下：

》 [k， poles] = rlocfind （num， den， p）

命令中输入参数 p 是指定的极点矢量。在控制系统分析过程中，常常需要求取对应某一极点附近的参数。假设求系统 $G(s)$中极点位置为–0.5 和–0.6 所对应的根轨迹增益及所有其它闭环极点，就可使用如下命令求得：

》 ng=1； dg= [1 3 2 0]；

》 [k, clpoles] =rlocfind （ng， dg， [-0.5, -0.6]）

则它的输出为

```
k =
     0.3750        0.3360
```

```
clpoles
     -2.1514      -2.1381
     -0.5000      -0.6000
     -0.3486      -0.2619
```

3.4 传递函数的常用命令

本节介绍一些常用于对传递函数进行分析与变换的命令。

printsys 命令是传递函数显示命令。其格式如下：

```
》 printsys （ num， den ）
```

例如：

```
》 ng= [ 1 1 ]； dg= [ 1 3 20 ]；
》 printsys （ ng， dg ）
   num /den =
```

$$\frac{s+1}{s\text{^}3+3s\text{^}2+2s}$$

求传递函数的极点与零点有多种方法。例如，可以使用 roots 命令分别求得分子多项式与分母多项式的根；也可以使用 tf2zp 或者 pzmap 命令。tf2zp 命令格式如下：

```
》 [ z， p， k ] =tf2zp(num， den)
```

该命令可以得到零点列矢量、极点列矢量与增益常量。该命令的逆命令为 zp2tf，它将用已知的零点与极点建立一个传递函数。

pzmap 的命令格式如下：

```
》 [ p， z ] =pzmap （ num， den ）
```

如果该命令中没有输出变量，则执行该命令后将会得到绘制好的系统零极点图。该命令也可以用于绘制已知的极点（列矢量）与零点（列矢量）图形。

当一个传递函数不是互质的（即有互相可以抵消的零、极点）时，可以使用 minreal 命令抵消它们的公共项而得到一个较低阶的模型，其命令格式如下：

```
》 [ numr ,  denr ] = minreal ( num, den, tol )
```

命令中第三个输入参数是（可选的）容差。当零极点不是完全相等，但是却非常接近时，我们仍然可以通过改变容差的大小，强迫让它们抵消掉。

最常用的对传递函数进行变换的命令为传递函数的乘、加与反馈连接命令。系统框图可以使用 SIMULINK 命令（见后面）进行分析和仿真。简单的框图分析可以使用 series 、 parallel 、feedback 与 cloop 命令，采用传递函数的形式进行分析与处理。这些命令的格式如下：

```
》 [ nums, dens ] = series ( num1, den1, num2, den2 )
》 [ nump, denp ] = parallel ( num1, den1, num2, den2 )
》 [ numf, denf ] = feedback(num1, den1, num2, den2, sign)
》 [ numc, denc ] = cloop ( num, den, sign )
%对应于单位反馈系统。
```

每一条命令分别对应的情况如下。

串联：　　$G_s(s)=G_1(s)G_2(s)$

并联：　　$G_p(s)=G_1(s)+G_2(s)$

反馈：

$$G_f(s)=\frac{G(s)}{1+G_1(s)G_2(s)}$$

单位反馈：

$$G_c(s)=\frac{G(s)}{1+G(s)}$$

sign 是可选参数， sign=-1 为负反馈，而 sign=1 对应为正反馈。缺省值为负反馈。

3.5 控制系统分析例题

【例 3-1】使用本章所学的 MATLAB 命令，对下面的系统进行分析。系统传递函数如下：

$$G(s)=\frac{1}{s(s+1)(s+2)}\text{，}\quad K=1.5$$

其结构图见图 3-1 。

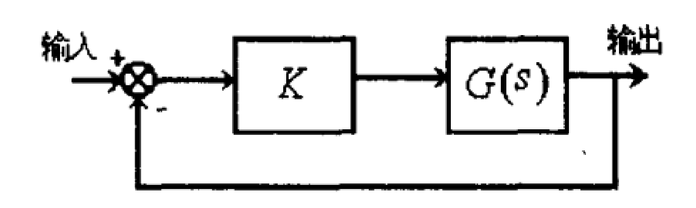


图 3-1　系统框图

频率响应分析：首先可以写出系统的开环传递函数，得到对应 K=1.5 时的系统频率特性伯德图。使用 logspace 命令生成具有 100 个频率点的对数坐标轴。

```
》 K=1.5； ng=1； dg=poly( [ 0 -1 -2 ] )；
》 w = logspace （-1， 1， 100）'；
》 [ m， p ] =bode （ k*ng， dg， w ）；
》 subplot(211)； semilogx （ w， 20*log10(m)）；
》 subplot(212)；semilogx （ w， p ）
```

由其绘制的伯德图（见图 3-2）可见，系统的增益裕量与相角裕量大约为 10dB 与 45°，也可以使用 margin 命令验证这一点。

```
》 [ gm， pm， wpc， wgc ] =margin （ m,p,ω）
     gm =
          4.002
     pm =
          41.5332
    wpc =
          1.4142
     wgc =
          0.6118
```

gm 值表示系统增益再增加 gm(=4)，系统就由稳定开始转入不稳定。这个值折合为 12dB 。

下面介绍我们如何求得带有单位圆的奈魁斯特图。为生成一个单位圆，使用 linspace 命令建立了一个 0 ～ 2 π 的空间矢量。这个单位圆是通过在复平面上绘制 $e^{j\omega}$ 的虚部与实部的轨迹而得到的。

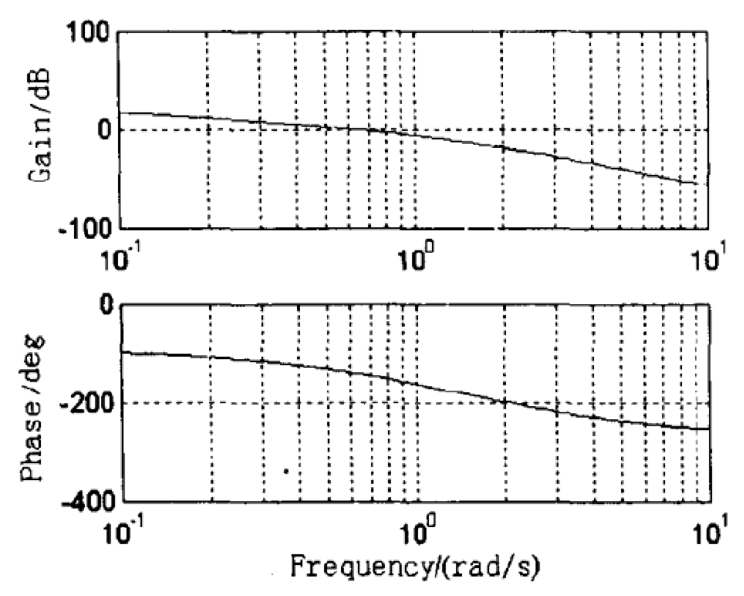


图 3-2　系统 *KG*（*s*）的幅频与相频图

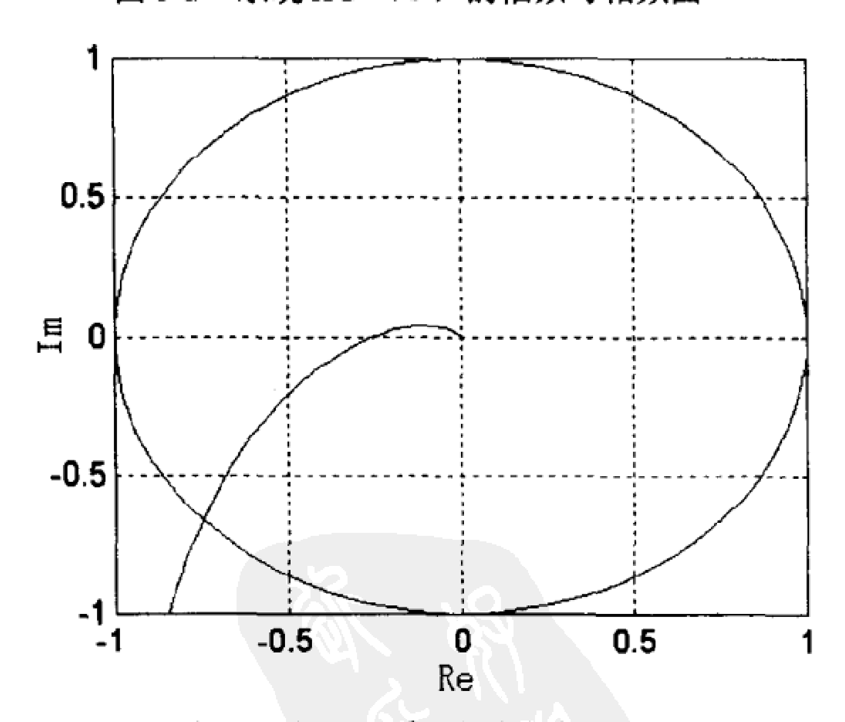


图 3-3　例 3-1 系统的奈魁斯特图

```
》 w2 = linspace(0,2*pi,100)';  ejw=exp(j*w2);
》 r2=real(ejw);  i2=imag(ejw);
》 [ r , i ] =nyquist(k*ng,dg,w);
```

》 axis('square'); plot(r2, i2); axis

下面一行的命令可绘制带有单位圆的奈魁斯特图。

》 plot(r2, i2, r, i); grid;

因为奈魁斯特图中的轨迹没有包围(-1,j0)点，所以图 3-3 中所示的系统是稳定的。注意在例题图 3-4 中的增益裕量为 12dB 。

》 nichols(k*ng , dg , w)或 plot(p , 20*log10(m))

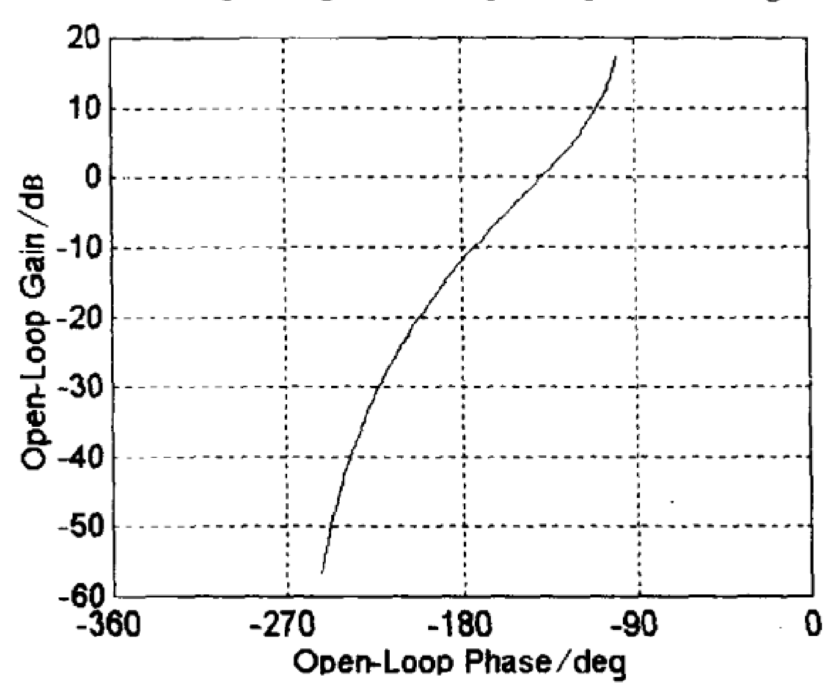


图 3-4 $KG(s)$的尼克尔斯图

从闭环系统的伯德图中，可以得到系统带宽及谐振峰值等特征参数。设 $T(s)$为系统闭环传递函数，使用 cloop 命令可求得系统的闭环传递函数：

》 [nt , dt] =cloop(k*ng, dg);

》 mc=bode(nt, dt, w); subplot(211),

》 semilogx(w, 20*log10(mc)); grid

由图 3-5 我们可以看到闭环系统谐振峰值是 3dB，而带宽约为 1rad/s 。

根轨迹分析法：下面将使用根轨迹法对于不同的增益的变化预指系统响应。

输入

》 rlocus(k*ng , dg)

可以得到系统根轨迹图，见图 3-6 。

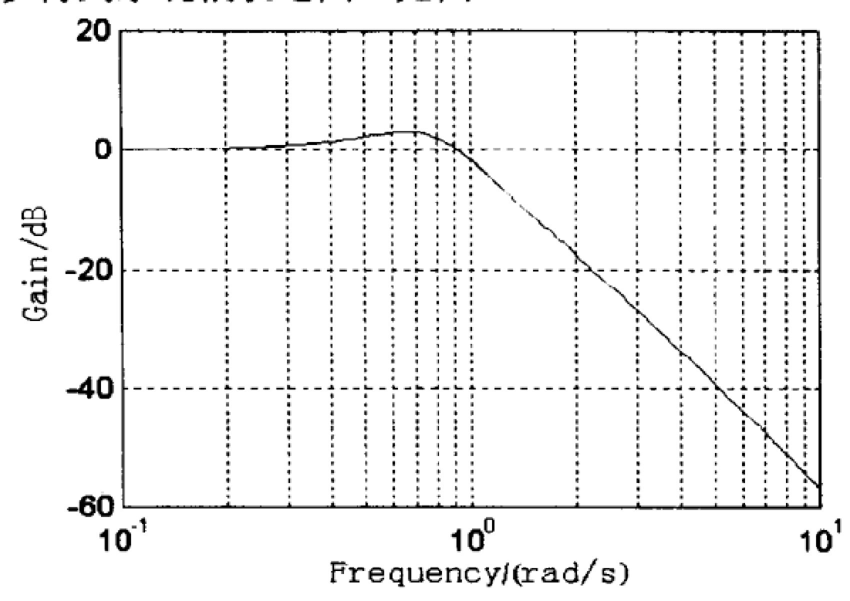


图 3-5　闭环系统的伯德幅值图

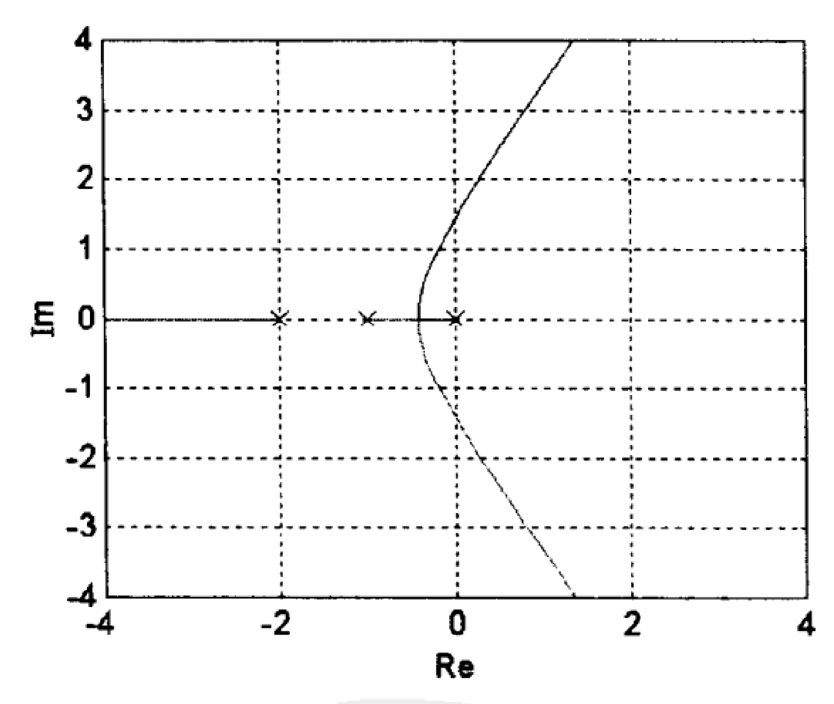


图 3-6　系统根轨迹图

使用 rlocfind 命令,就可以知道当 $K>6$ 时，系统变成不稳定系统。还可以通过求得该临界点（K=6）附近一定范围内极点的参数得到其准确值。

```
》 clpole=rlocus（ng， dg, [ 0.3:0.1:7 ]）
》 range = [ 0.3:0.1:7 ]'; [ range， clpole ]
ans =
```

```
0.3000    -2.1254    -0.6611                 -0.2135
0.4000    -2.1597    -0.4201-0.0932i         -0.4201+0.0932i
5.9000    -2.9909    -0.0046-1.4045i         -0.0046+1.4045i
6.0000    -3.0000     0.0000-1.4142i          0.0000+1.4142i
6.1000    -3.0090     0.0045-1.4238i          0.0045+1.4238i
```

上述数据显示了该区域内增益及对应的闭环极点位置。由此可得出如下结论：

(1) $0 < K < 0.4$ 时，闭环系统具有不同的实数极点，表明系统处于过阻尼状态。

(2) K=0.4 时，对应为分离点，系统处于临界阻尼状态。

(3) $0.4 < K < 6$ 时，系统主导极点为共轭复数极点，系统为欠阻尼状态。

(4) K=6 时，系统有一对虚根，系统处于临界稳定状态。

(5) $K > 6$ 时，系统出现右根，系统处于不稳定状态。

为了验证上述结论，我们将求得闭环系统的传递函数，并求得五个系统的阶跃响应。K值分别为 0.25 、0.4 、1.5 、6 和 8 。MATLAB 程序 3-1 为求得 5 个系统的阶跃响应程序清单。

```
% MATLAB PROGRAM 3-1
k=1.5; ng=1; dg=poly([0 -1 -2]);
rangek=[0.25 0.4 1.5 6 8]; t=[0: 0.2: 20]';
for j=1:5
[ntc, dtc]=cloop(ng*rangek(j), dg);
y(:,j)=step(ntc, dtc, t); end
subplot(211), plot(t, y(:,1:3)), grid
subplot(212), plot(t, y(:,4:5)), grid
gtext('k=0.25'), gtext('k=0.4'), gtext('k=1.5'),
gtext('k=6'), gtext('k=8')
```

这五种系统的阶跃响应（三个是稳定的，两个是不稳定的）见图 3-7 。

下面着重讨论欠阻尼即 K=1.5 时情况。

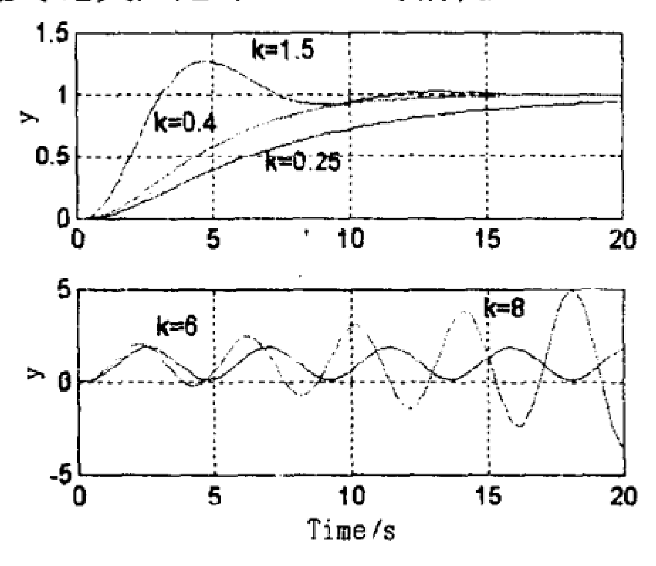


图 3-7 不同 K 值的系统阶跃响应

系统阶跃响应的超调量：即峰值与稳态值之差，且与系统稳态值之比。超调量计算过程如下：

```
》 y3=y(:, 3); mp=max(y3); yss=y3(length(t));
》 pos=100*(mp-yss)/yss
pos=
      27.9035
```

稳态误差分析：我们知道，单位反馈系统在零点处有一极点的系统为典型 I 型系统，它的稳态误差 e_{ss} 为零。对于斜坡响应的误差为 $1/K_v$（此处 K_v 为速度误差系数）。

$$K_v = \lim_{s \to 0} sG(s) = 1.5/2 = 0.75$$

因此，e_{ss}=1.33 。

下面使用 lsim 命令去求斜坡输入响应，并且证明所得到的结果（见图 3-8 ）。计算斜坡输入响应的简单办法是在系统闭环传递函数上再乘一个 $1/s$ ，MATLAB 程序 3-2 为计算系统斜坡响应的程序清单。

```
% MATLAB PROGRAM 3-2
k=1.5 ; ng=1 ; dg=poly([0 -1 -2]);
[nt , dt]=cloop(k*ng , dg);
dtt=[dt , 0]; t2=[0 : 0.1 : 10]';
```

```
yramp = step(nt , dtt , t2);
subplot(211),plot(t2 , [t2 yramp]), grid
subplot(212),ess=t2-yramp ; plot(t2,ess),grid,
ess(length(ess))
```

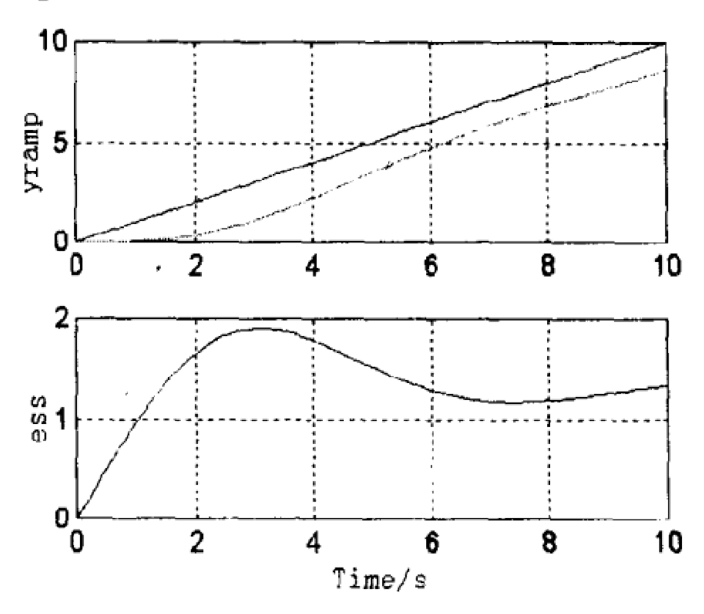


图 3-8 系统斜坡响应及误差

由下列命令可以验证其稳态误差：

```
》 ess=t2-yramp ; subplot(212), plot(t2,ess),
》 grid , ess(length(ess))
ans=
      1.3307
```

滤波器性质：由闭环系统的伯德图（见图 3-5 ）可以看出，系统特性为一低通滤波特性。大多数系统都具有低通滤波特性，这意味着系统对高频信号具有很大的衰减特性，可通过求得随机噪声信号在欠阻尼系统中的时间响应来证明这一点。 **rand** 命令用于将一噪声信号加入到一阶跃输入信号中，而 **lsim** 命令用于求解其响应。

```
》 noise=ones(t2)+rand(t2);
》 ynoise=lsim(nt,dt,noise,t2);
》 subplot(211),plot(t2, [ noise yniose ] )
```

式中， t2 是一个矢量， ones(t2)为一具有相同元素数的‘1’矢

量，而 rand(t2)建立另外一个具有相同大小、元素在 0 ～ 1 之间的随机数的矢量。两者之和构成了一个噪声阶跃输入信号。为了便于比较，可将输入与输出绘制在同一图中，见图 3-9 。正如预料的结果一样，系统对高频噪声信号有一个很大的衰减作用。由于噪声信号具有 0.5 单位的幅值，因此其输出信号中出现了一个 0.5 单位的偏移。通常噪声信号的幅值相对于其输入信号的幅值要小得多，因此其作用几乎可以忽略。

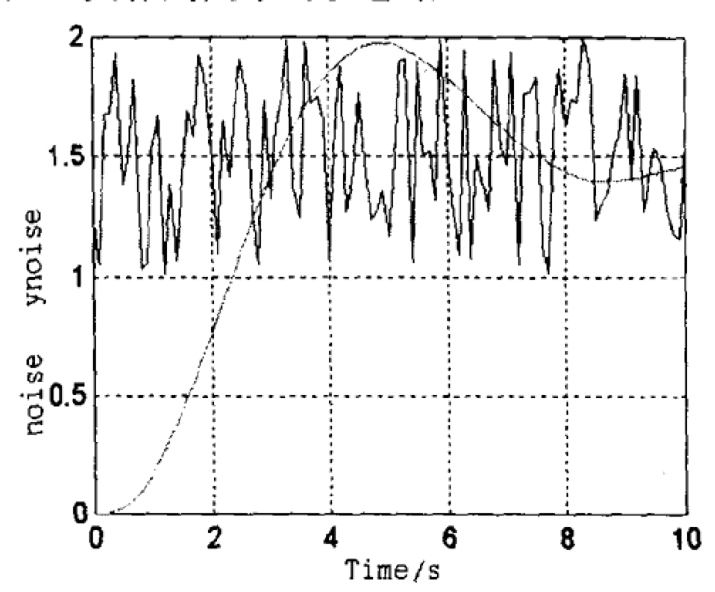


图 3-9 随机噪声阶跃输入的系统响应

【例 3-2】 延时作用。

延时环节应用于许多控制系统，如过程控制与制造系统等。在线性时不变连续系统中，用 e^{-st} 表示延时。注意采用频率响应的方法可以较方便地分析系统特性。

$$e^{-j\omega T}=1\angle -\omega T$$

因此，延时环节将保持输入幅值不变，而简单地加入滞后的相角。带有延时的相角滞后会使系统趋于不稳定。如果采用伯德图分析方法，则只需要在系统的相频响应中减去延时的相角，即可得到其系统的伯德图。

设：

$$G(s)=\frac{2}{s+1}\qquad T=1$$

绘制系统伯德图的 MATLAB 程序如下：

```
% MATLAB PROGRAM 3-3
n=2； d=[1 1]； del=1；
ω = logspace(-2， 0.7， 100)'； [m, p]=bode(n, d, ω)；
pd=p-(del*ω*180/pi)； % 减去滞后相角
subplot(211)， semilogx(ω， 20*log10(m))， grid；
subplot(212)， semilogx(ω， [p pd])， grid
```

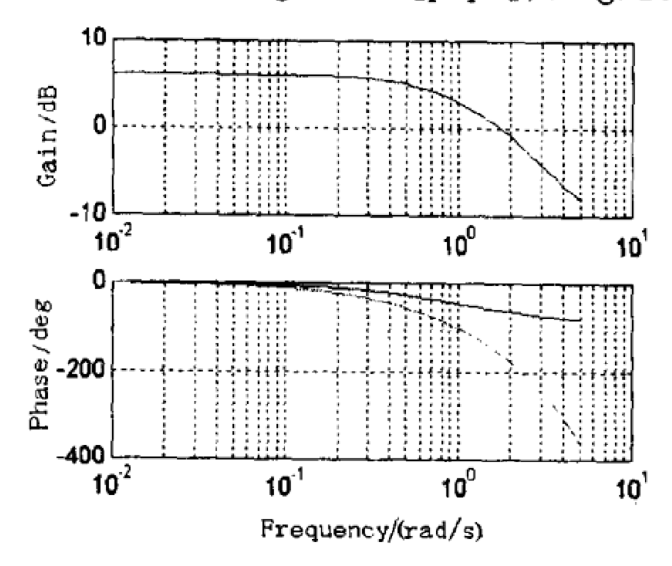


图 3-10 系统加入延时环节的作用

由图 3-10 可见，原系统具有无穷大的增益裕量与 120°的相角裕量。加入延时环节后，增益裕量变为 2dB，而相角裕量变为 30°。由此可见，延时环节的作用对系统影响还是很大的。

延时环节的奈魁斯特图可以由下式得到：

$$G(j\omega)e^{-j\omega T}=[R(\omega)+jI(\omega)]e^{-j\omega T}$$

式中， $R(\omega)=\mathrm{Re}[G(j\omega)]$ ； $I(\omega)=\mathrm{Im}[G(j\omega)]$

因此，可以计算出带有延时环节的系统的实部与虚部的频率响应。MATLAB 程序 3-4 为绘制其奈魁斯特图的程序清单。

```
% MATLAB PROGRAM 3-4
n=2； d=[1 1]； del=1；
ω2=logspace(-2, 1.3, 200)'； [r, i]=nyquist(n, d, w2)；
rd=real((r+j*i).*exp(-j*w2*del))；
id=imag((r+j*i).*exp(-j*w2*del))；
```

```
plot([r, rd], [i,id]), grid
```

注意 real 、 imag 与 exp 命令的用法。系统奈魁斯特图见图 3-11，由图可见，系统仅有很小的稳定裕量。

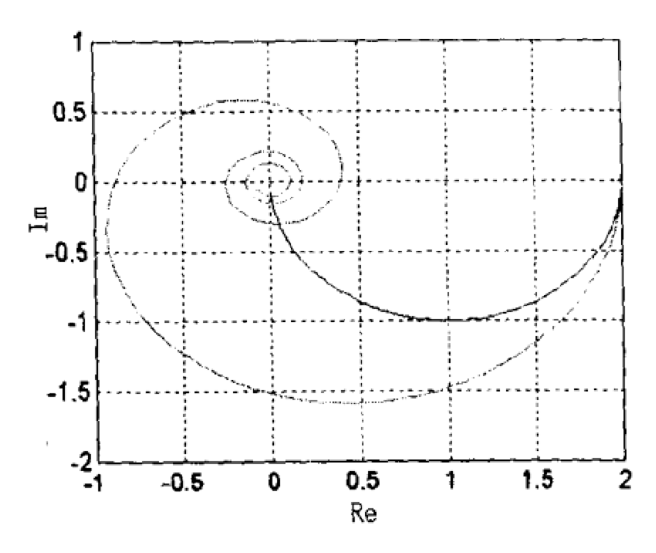


图 3-11 带有延时环节的系统的奈魁斯特图

3.6 SIMULINK 简介

SIMULINK 是 MATLAB 软件的扩展，它主要用于动态系统的仿真。它与 MATLAB 的主要区别在于，它与用户交互接口是基于 Windows 的图形（使用方块图）编程方法。在 SIMULINK 中也加入了许多 MATLAB 函数功能。由于 SIMULINK 是一个具有许多功能的程序，在此不可能用较短的章节来描述其全部功能。下面将简要地介绍其基本内容、基本概念与在仿真过程中常用的一些命令。通过一个实例，介绍如何在 SIMULINK 环境下建立一个系统模型。在上机完成一个实际系统分析之后，应该能够掌握其基本使用的分析方法去解决一些实际问题。在例题中，给出了其方块图模型的建立方法及其仿真结果。通过对这些例题的讨论，应该达到如下两个目的：其一，通过例题了解实际简单系统模型的建立；其二，使用 SIMULINK 来讨论分析一些系统设计的基本概念（如饱和效应与不稳定零极点对消的作用等）。

在本书中，假定大家都熟悉 Windows 的操作功能，知道如

何使用鼠标器，什么是单击与双击鼠标器，拖曳等用法。在 SIMULINK 操作过程中，我们会频繁地操作鼠标器。在 PC 机上鼠标器的操作主要是左键（如果您的鼠标器有三个键，则中间键对该软件是没有用处的）。由于 SIMULINK 的特点，在本节的学习过程中，应该使用你的计算机，通过上机练习来学习 SIMULINK 。

3.6.1 建立模型的基本步骤

第一步是如何进入 SIMULINK 。只要你在 MATLAB 提示符下敲入 SIMULINK ，按回车键即可启动 SIMULINK 软件。在启动 SIMULINK 之后， SIMULINK 的主要方块图库将显示在一个新的 Windows 中。建立方块图所需的大部分类型的方块都在这个 Windows 中。你所要做的工作只是选择你所需方块的种类，将它们联接起来，并设定每一个方块内的参数，这是非常简单的工作。主方块库被分成不同的类别,见图 3-12 。

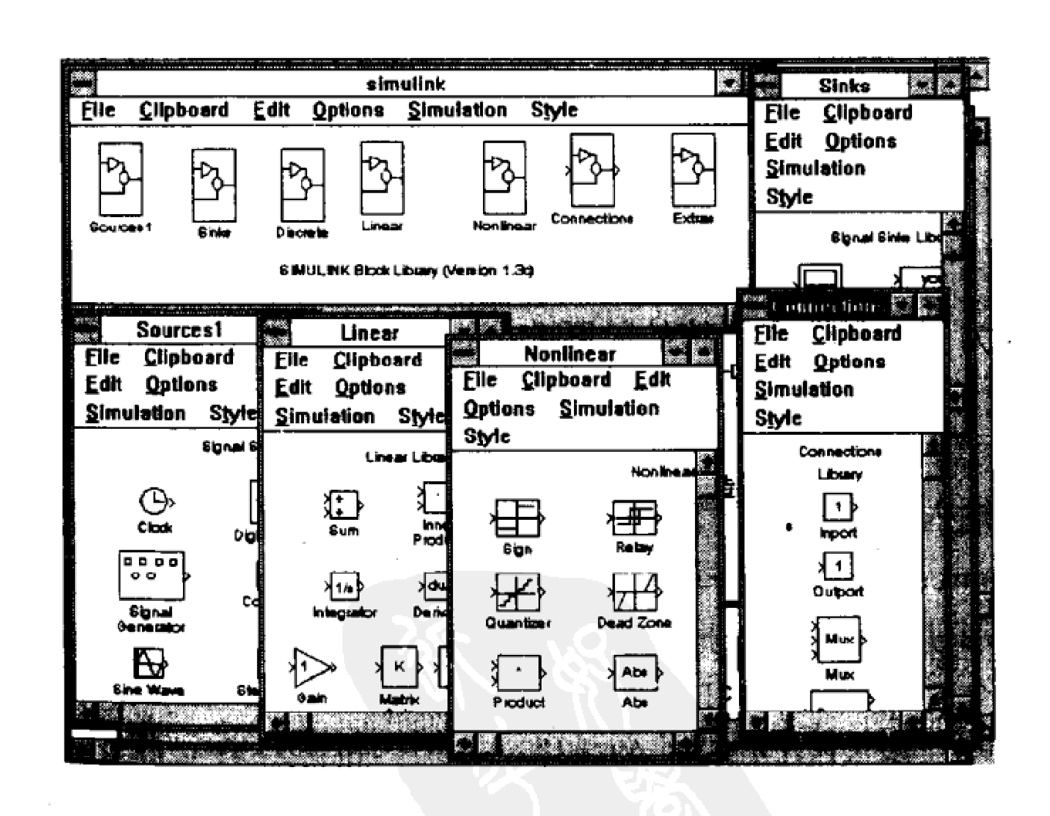


图 3-12 SIMULINK 方块图库与其中的部分内容

为了看清其中每一类别的内容，双敲那个类别的图形，一个

新的窗口会被打开。在图 3-12 中，已经打开五个窗口，并显示其中每一窗口的内容。

当打开 SIMULINK 窗口后，菜单标题也会发生变化。文件(File)、编辑(Edit)及式样(Style)是在 Windows 环境下的常用菜单项。后面将对此做一简单的介绍。在建模之前，你需要创建一个工作区域。创建一个工作区域的方法为，选择 File 项，然后再选择 New,这将开始一个新的窗口，其窗口名为“ Untitled1 ”，可以在该窗口内构造系统模型，并称这个窗口为工作窗口。

【例 3-3】以具有双积分环节的系统 $G(s)$为例。该系统开环是不稳定的，为了使系统稳定，使用超前校正环节 $K(s)$进行串联校正，见图 3-13 。

$$G(s)=\frac{1}{s^2} \qquad K(s)=\frac{(s+1)}{(s+5)}$$

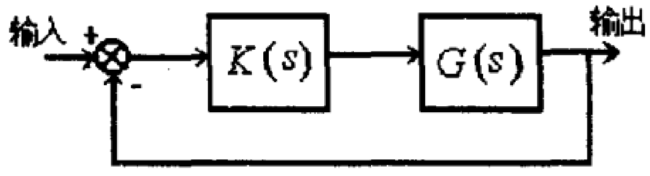


图 3-13 例 3-1 系统结构框图

为了得到这个系统的阶跃响应，可以由两个传递函数、一个求和点、一个输入源及两个输出观测点等 6 个部分组成这个系统。

输入源的元件位于 Sources 库。双击这个库的图标将打开这个 Sources 窗口，选择其中的 StepFcn 源。其方法是，将所选择的图形使用鼠标器“拖曳”到工作窗口即可。这样等于拷贝了另外一个 StepFcn 源到工作窗口，而 Sources library 中的 StepFcn 依然保留。在 SIMULINK 的建模过程中，可能会打开许多窗口。当使用完毕后，建议将不再使用的窗口关闭，以便使屏幕清晰。

传递函数与综合点方块都位于线性部分库中。用同样方法，可将该库中的 Transfer Fcn 与 Sum 图形拖曳到工作空间，然后关

闭 Linear 库。

使用一陷点可以得到其仿真的输出结果，在 Sinks 库中有三个功能方块可用于显示或存储输出结果。Scope 功能块可以像一台示波器，实时地显示任何信号的仿真结果。To Workspace 功能块可以把输出值以矢量的形式存储在 MATLAB 工作空间中，这样可以在 MATLAB 环境下分析与绘制其输出的结果。To File 功能块可以把数据存储到一个给定名字的文件中。用同样方法，将 Scope 拖曳到工作空间，并关闭 Sinks 库窗口。

现在工作空间中已有六个功能块。下一步是确定每一功能块的参数。双击任何一个功能块，即可打开该功能块，且显示那个功能块的参数设定窗口，窗口中都附带有简要的说明，当然也可以使用 Help 选项，以得到更详细的参数设定说明。大多数功能块都带有其缺省值，这些缺省值是可以改变的。

打开其中一个传递函数功能块后可以见到，有两项空白参数分子项(Numerator)与分母项(Denominator)可填写。填入[10 10]和[1 5]在对应的方框处。当然这是在 MATLAB 中一般的矢量输入格式。输入完毕后，选择 OK 按钮（即将光标置于 OK 处，按鼠标器左键），即完成了该功能块的参数选择。然后再打开另外一个传递函数功能块，用同样方法，分别输入[1]和[1,0,0]，再选择 OK 按钮，完成参数选择，并关闭该窗口。

打开 Sum 功能块，在 List of Signs 处输入“+”、“-”符号。如果综合点超过了两个输入点，只要简单地输入其正、负号，即可自动地增加其相应的输入点。

打开 StepFcn 功能块，有三个空白框可以填入参数。Steptime 是阶跃响应的初始时间。此项可填 0 ，即零时刻开始阶跃响应。另外两项为初始值(Initial value)和终值(Final value)。这两项可分别输入 0 和 1 。

打开工作空间功能块。输入 y 作为变量名(Variable name)，对应最大行数项(Maximum number of rows)，输入 100 。每一行对应一个时间间隔。在系统仿真过程中，可以输入 0 到 9.9 ，间

隔为 0.1，生成 100 个点。

最后，要将这些方块连接起来。除 Sources 与 Sinks 功能块外，所有其它种方块中至少有一个输出点，即在方块旁有一个>符号指向外面，也至少有一个输入点，即在方块旁有一个>符号指向里面。 Sources 功能块没有输入点，只有输出点，而 Sinks 功能块没有输出点，因此它仅有一个输入点。

两个功能块连接方法如下：将鼠标器光标指向一个功能块的输出点，按下左键拖鼠标器的光标至另一功能块的输入点，然后释放左键，此时>符号点消失，而在两个功能块之间出现一条带有箭头的连线。

使用上述方法，连接阶跃输入到综合功能块的正输入点。在绘制过程中，应注意到，当拖鼠标器光标接近其正输入点时，该线会自动地与该输入口突然连接在一起。如果连线出现错误，可以使用鼠标器左键选中该线，然后再使用 Edit 中的 cut 命令将该线删除掉。重复上述过程， 将综合块与第一个传递函数块连接；第一个传递函数块与第二个传递函数块连接；第二个传递函数块连接到输出点(To Workspace)。

最后将反馈环连接，即从输出点到综合块的负输入点。此时，被控对象($1/s^2$)输出点与输出块连接，因此再没有>符号点。然而，可以从一个方块中引出多个输出。其方法是用鼠标器在被控方块输出点处单击并拖曳鼠标。在接近另外一个方块之前释放左键，此时可以得到另外一个带有>符号的输出点。可以使用这个新建立的输出点与综合点的负输入口相连。使用上述方法，可以修改连线，完成系统方块图的绘制。

最后一步是仿真(Simulation)，可以通过选择仿真菜单(Simulation Menu)执行仿真命令。有两个可以供选择的项： Start（开始执行）与 Parameters（参数选择）。在参数选择项中，可以有几种积分算法供选择。对于线性系统，可以选择 Linsim 算法。对应项分别输入如下参数：

Start Time　　　0　　　(开始时间)

Stop Time	9.9	(停止时间)
Relative Error	0.001	(积分一步的相对误差)
Minimum Step Size	0.1	(最小步长)
Maximum Step Size	0.1	(最大步长)

在 Return Variable 方框中，还可以输入要返回的变量参数。如在此方框中填入 t，在仿真之后可以在 MATLAB 工作空间中得到两个变量，即 t 与 y。参数选择完毕后，选择 OK 键，关闭该窗口。

此时你可以选择 Start 启动仿真过程，在仿真结束时，计算机会用声音给予提示。

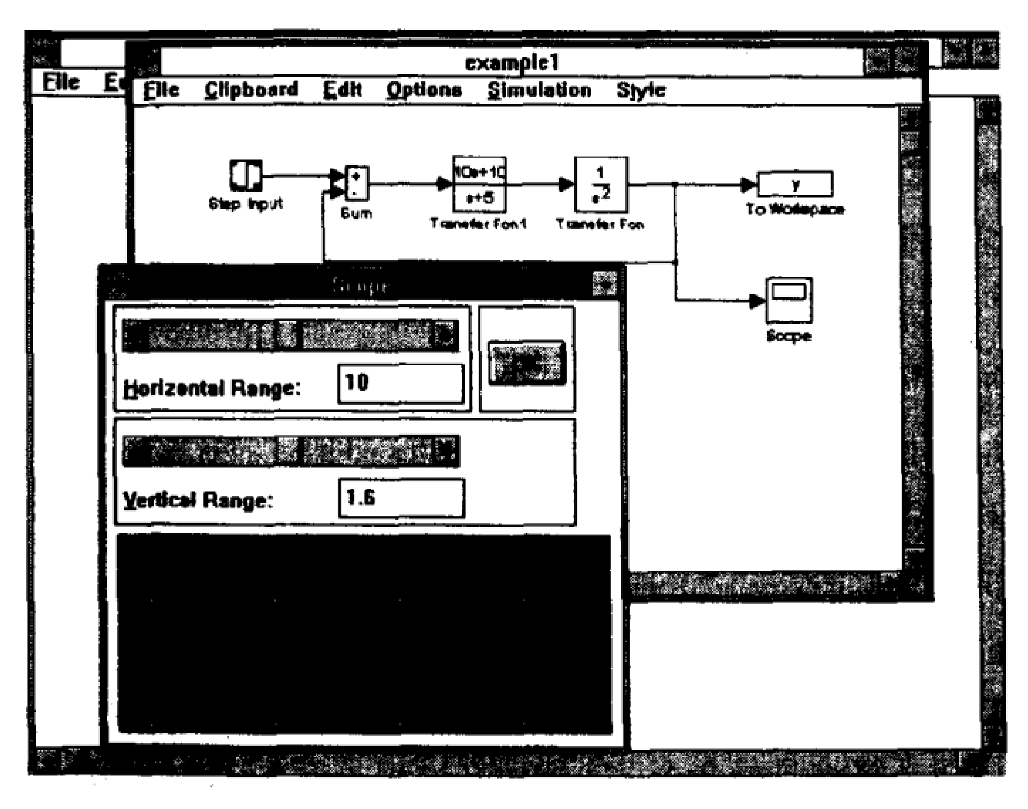


图 3-14　例 3-1 的系统方块图与其输出波形

我们可以通过 Scope （示波器）观察系统的仿真结果输出，可以把 Scope 连接到任何你想观测的点，调整好 Scope 的扫描量程(Horizontal Range)与显示的幅值量程（ Vertical Range ），同时也应调整 Scope 的窗口大小及位置，以利于观察系统的仿真过程。在此例中， Vertical Range 为 1.6， Horizontal Range 为 10 。

启动(Start)仿真程序，其输出结果见图 3-14 。

使用 Save 命令（在 File 菜单中）可以存储这个方块图（以“ example1 ”为文件名）。这样就建立了一个 MATLAB 的 M 文件，该文件描述这个系统的方块图。在 MATLAB 提示符下，只要输入该文件名，就可以得到该系统的方块图。

当你执行一次仿真过程后，即可在 MATLAB 工作空间中得到两个变量 t 与 y 。使用 Plot 命令绘制 y 对应 t 的曲线，即可得到如图 3-15 所示的结果。

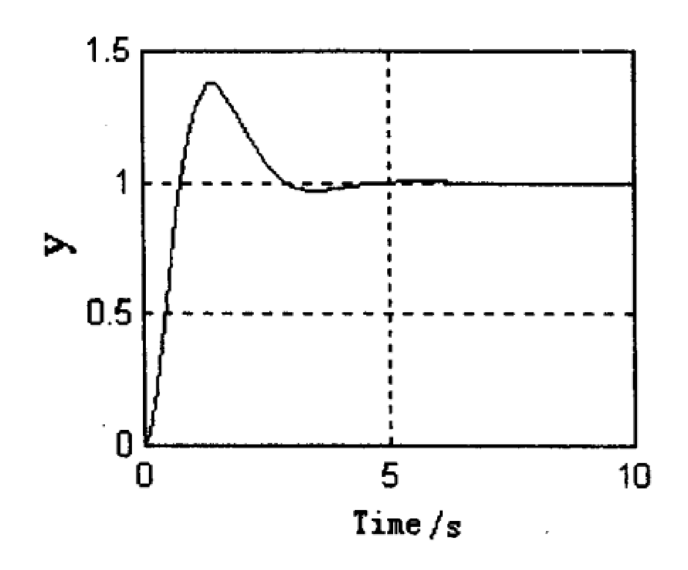


图 3-15　例.3-1 的系统阶跃响应

3.6.2　SIMULINK 命令

1.命令行中的SIMULINK命令　在MATLAB环境下,可以直接使用仿真算法各种命令，实现仿真过程。所有算法的命令格式都是一样的，请见下列的命令格式：

```
》[t,x,y]=linsim('model',tf,xo,option,u)
```

输入参数如下：

第一个参数 'model' 为 SIMULINK 生成模型名；

第二个参数为仿真的终了时间值。该参数也可输入两个元素的行矢量。即其中一个元素表示非零初始时间值；

其余的输入参数是可选的。第三个参数是初始条件矢量。这

些参数值将替代已在方块图中设置的初始条件值。此外，在命令行还可以仿真带有初始条件的传递函数（该传递函数在 SIMULINK 中的参数设定窗口中没有任何初始条件设定栏）；第四个参数是一个可选的控制积分特性的矢量，其中包括步长与容许误差。

最后的输入参数为外部输入信号，可以在 SIMULINK 的信号源库中选择输入，也可以直接从命令行中设定输入。对于命令行设定输入情况，有字符串或数值表可供选择。对于字符串，任何有效的 MATLAB 函数与表达式都是允许的。例如："sin"，MATLAB 函数 sin 将对应时间 t 进行计算，并用于外部输入。如果输入是一个数值表，则第一列一定是一个增序的时间矢量；另外列可以是输入矢量。例如，这个系统有 3 个输入，矢量 $\boldsymbol{u}$ 将有四列。

输出参数是时间矢量、状态与输出矢量。如果命令中不含左参数，将自动地绘制状态变量图。如果该模型已经有输出，则输出也将被绘制在图上。例如，下述命令将完成例 3-3 的系统仿真。

```
》 linsim ( 'examplel',10)
》 euler( 'examplel',10,[1 2 3])
```

第一个命令使用 linsim 算法；第二个命令使用 euler 算法，并带有特定的初始条件。

2. 积分算法的选择　在 SIMULINK 中，有 6 种可供选择的积分算法，可根据不同的问题，选择不同的算法。一般地说，选择原则如下：

(1) linsim 算法：主要用于线性系统，或系统中仅含几个非线性元件的情况。该算法适用于既有较大极点也有较小极点的系统(stiff system)的仿真运算。

(2) rk45, rk23 ：龙格-库塔五阶与三阶算法。这两种算法适用于高阶非线性与不连续系统，也适用于连续与离散的混合系统。rk45 算法比 rk23 算法要快而且更准确。而 rk23 算法取用较少的计算点，因此导致其绘制图形不平滑。该种算法不适用于 stiff

system 的情况。

(3) gear 算法： gear 为 Gear 的预测-补偿方法。该算法适用于平滑的非线性系统。该算法也特别适用于 stiff system，但该算法不适用于不连续的系统。

(4) adams 算法： adams 是亚当预测-补偿方法。它适用于平滑的、非线性与非 stiff system 的系统。

(5) euler 算法： euler 是欧拉方法。它是最简单的数值算法，也是最快的算法， 但是它的准确性与稳定性较差。例如,对于一个稳定系统的运算，若运算的步长选择不适当，很可能得到一个不稳定的响应结果。该方法经常用于快速验证采用其它方法的运算结果。

3.线性化问题　考虑下列非线性系统：

$$\dot{x} = f(x,u,t)$$

$$y = g(x,u,t)$$

上述非线性系统可以在均衡点（x^*， u^*）附近实现线性化。定义扰动状态与控制输入如下：

$$\delta x = x - x^* \qquad \delta u = u - u^*$$

对于一阶近似，可使用 Taylor 扩展级数,线性方程给定如下：

$$\delta' x = [\frac{\partial f}{\partial x}]\delta x + [\frac{\partial f}{\partial u}]\delta u$$

$$y = [\frac{\partial g}{\partial x}]\delta x + [\frac{\partial g}{\partial u}]\delta u$$

式中，$\partial f/\partial x$ 表示矢量函数 f 对应状态矢量 x 的函数（雅各比）行列式。函数（雅各比）行列式是在特殊平衡点的（ x^*， u^*）所有计算值。如果该系统不是一个时间函数（时变系统），雅各比矩阵将为一常数矩阵。

linmod 命令方式用于寻求线性系统的状态空间表达式。其命令格式如下：

```
》[a， b， c， d]=linmod( 'model')
》[a， b， c， d]=linmod( 'model' ， x， u)
```

式中，model 是 SIMULINK 生成的方块图模型的名字。在第二种命令格式中，对应于状态与控制输入的平衡点可以设定。该命令也有其它形式的格式，允许控制状态变量、控制变量等的波动范围。该种命令的离散形格式是 dlinmod，它可以用于连续与离散混合的时变系统。对于线性系统，该命令还可给出状态空间方程。

以本节例 3-3 中的系统为例，使用上述命令会求得系统的闭环传递函数。在做这项工作前需要在系统方块图中再加入两个功能块。这些功能块是 Import 与 Outport，它们位于 Connections 库中。这些块是与外部输入与输出交流的通道。如果没有这些功能块，linmod 命令将给出零矢量作为输入与输出的连接矩阵。修改后的方块图见图 3-16。修改后仍然以同样的文件名存盘。

求取系统传递函数的 MATLAB 命令如下：

```
》[a， b， c， d]=linmod( 'examplel');
》[nt， dt]=ss2tf(a， b， c， d)
nt=
    0     0          10.0000    10.0000
dt=
    1.0   5.0000     10.0000    10.0000
```

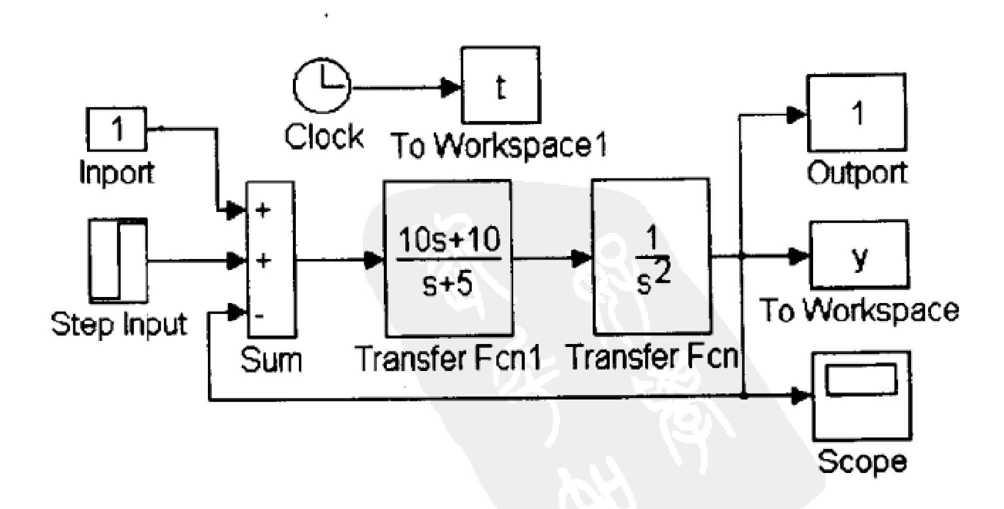


图 3-16　SIMULINK 系统结构框图

与此相关的trim命令是计算系统平衡点的命令。命令格式如下:

```
》[x , u , y , dx]=trim( 'model')
》[x , u , y , dx]=trim( 'model', x0, u0, y0, ix, iu, iy)
```

该命令可求得对应系统进入稳态(即状态变量的微分项为0)的状态变量与输入值。第二种命令格式形式有两套附加参数。第一套参数含有初始的期望值。第二套参数中含有一些可修改的元素目录号。例如,考虑如下系统:

$$\dot{x}_1 = -x_1 - x_2 + u$$
$$\dot{x}_2 = x_1 - 2x_2 + u$$
$$y = x_1 + 3x_2$$

试求输出值为1时,稳态的状态变量值与控制输入值。

使用trim命令,需要在SIMULINK环境下定义这个系统。执行simulink命令,打开一个新的文件窗口;然后再打开linear库,将State-space功能块拖曳到新的窗口;打开这个功能块置入它的参数;再打开Connections库将Inport与Outport功能块拖曳到新的工作空间;连接所有功能块,并以'trimtest'命名存盘。返回到MATLAB执行如下命令:

```
》[x, u, y, dx]=trim( 'trimtest' , [], [], 1, [], [], 1)
 x=
    0.1429
    0.2857
 u=
    0.4286
 y=
    1.0000
 dx=
    1.0e-016
    0.2776
    -0.2776
```

4.数值插补　SIMULINK 中的积分算法使用的是变步长方法，因此使得其仿真输出点之间不是等距离的。这样使得在不同条件下得到的仿真结果难以比较（例如：比较在不同初始条件下的响应）。在这种情况下，为了能够比较两者的结果，应该插入数值点，使得两个结果具有相同的输出点。MATLAB 的插值命令有 spline 、 table1 与 interp2 。

设矢量 y 为矢量 x 的函数，若给定另一个横坐标矢量 x_1，使用 interp2 命令可以通过插值法求得对应的纵坐标矢量 y_1。其命令格式如下：

```
》 y1=interp2(x,y,x1)
```

具体情况请见下面的例子。

3.6.3 例题

【例 3-4】我们将研究前面例子中的饱和效应。饱和特性在控制系统中是很普通的非线性现象，例如运算放大器特性、电动机的磁场饱和现象等。饱和环节几乎存在于每一个物理系统中。虽然饱和效应可以使用描述函数和其它近似的非线性方法进行分析，我们仍将使用仿真来分析它的作用。饱和环节的输入-输出特性见图 3-17 。

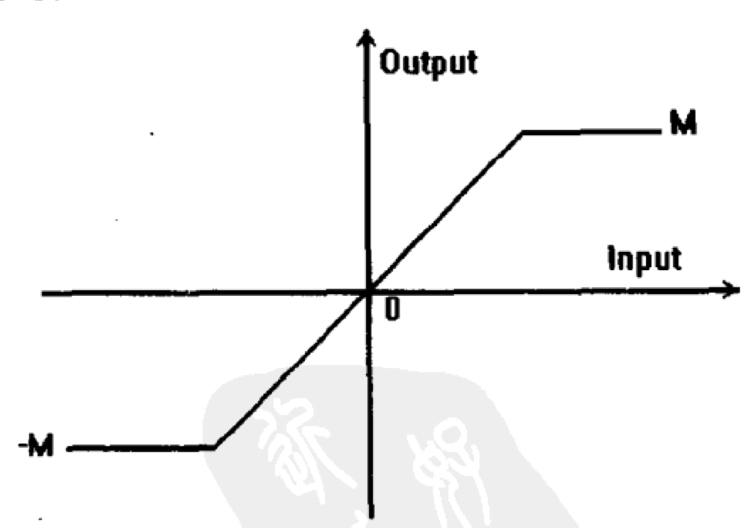


图 3-17　饱和环节的输入-输出特性

该非线性元件的增益为

$$k=\begin{cases} M/E & |\text{input}|<E \\ M & \text{input}>E \\ -M & \text{input}<E \end{cases}$$

总之，该放大器增益随着外部输入增加而减小。一般地讲，增益的减少使系统趋于稳定，使系统惰性增加。当然，对于某些条件稳定系统（如在低增益时不稳定的系统），饱和会使系统不稳定。事实上，在饱和期间，系统已暂时变为开环系统。因此，对于含有饱和非线性环节的系统，必须能够允许这种暂时的开环工作方式。

如图 3-18 所示，在生成系统结构框图之后存盘（以“ exz ”为文件名），为仿真工作做好准备。改变输入信号电平从 1 ～ 6，其系统饱合环节的上下限为± 1，我们将对这 6 种情况分别进行仿真并计算其超调量。最后，将其绘制在同一图中进行比较。

在系统仿真的程序中，将使用 for 命令。由于系统仿真将得到不同的数值点，因此全部的输出都使用 100 个点的时间轴矢量 t 进行插补。 MATLAB 程序 3-5 为系统仿真程序清单。

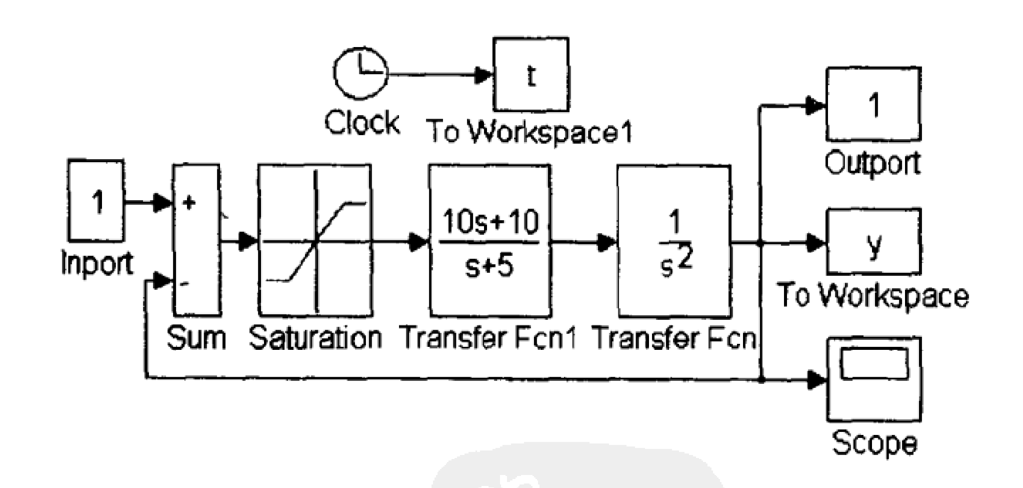


图 3-18　例 3-4 的系统结构框图

```
% MATLAB PROGRAM 3-5
t=[0:0.1:9.9]' ;
for i=1:6
```

```
ut=[t, i*ones(size(t))];
[tt,xx,yy]=linsim( 'example2',10,[],[],ut);
yint=interpl(tt,yy,t, 'spline'); y(:,i)=yint;
plot(tt,yy); grid , hold on,
i=i+1;
end
hold off , grid
sy=size(y); yss=y(sy(1),:);
pos=100*(max(y)-yss)./yss
pos=
    37.6548   34.6111   31.4223   29.5730
    29.7324   30.7281   30.7281
```

其仿真结果见图 3-19 。由图可见，随着输入电平的增加，过渡过程时间加长。我们知道，对于线性系统，当输入电平发生变化时，输出电平也随之发生变化，然而其动态特性曲线的特征参数不会发生变化。而在带有饱合环节的非线性系统中，其动态参数（如过渡过程时间与超调量）是随着输入电平的变化而改变的。这就是非线性系统的特点。

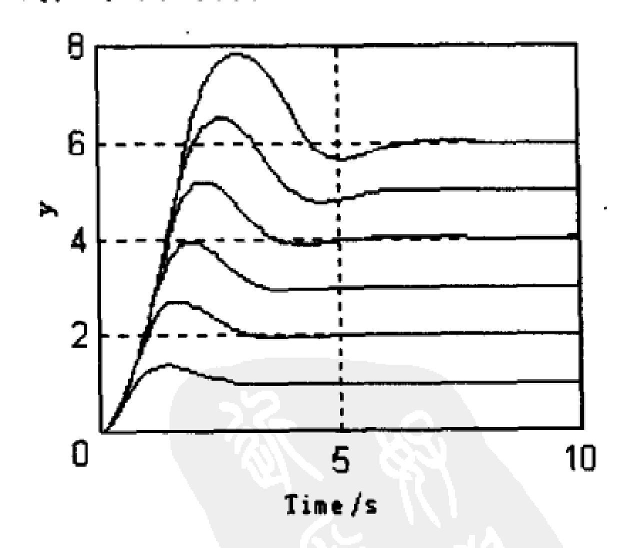


图 3-19 例 3-4 中带有饱合环节的系统阶跃响应

【例 3-5】零极点对消作用。

在本例中，将研究系统零极点对消的作用。稳定的零极点对

消是很普通的也是有效的设计策略。该方法可以称为极点配置方法，即用零点抵消掉稳定的极点，然后再将极点放置在理想的位置。它的设计思想是来源于陷波滤波器设计方法。尽管在实际系统中，完全准确的抵消是不可能的（这是由于任何系统或控制环节都存在着误差，对于数字系统也会由于字长的限制而出现误差），但是该方法还是很普通与有效的方法。本例题主要的目的是要通过分析与仿真，证明不稳定零极点的对消效果，并说明为什么不必这样做。以下面系统为例：

$$G(s)=\frac{1}{s-1} \qquad K(s)=\frac{s-1}{s+1}$$

式中； $G(s)$是一被控对象； $K(s)$为一补偿环节。当然由 $G(s)*K(s)$ 构成的系统在限定输入限定输出(BIBO)界限内是稳定的。由于系统是二阶的，因此可以用两个一阶微分方程来表示，并由此求其阶跃响应。

$$\dot{x}_1 = x_1 + u$$
$$\dot{x}_2 = -2x_1 - x_2$$
$$y = x_1 + x_2$$

SIMULINK 系统结构框图见图 3-20 。

设已知 $x_1(0)=\alpha$ ， $x_2(0)=\beta$，求解上述方程可得：

$$x_1(t)=(\alpha+1)e^t-1$$
$$x_2(t)=2+(\alpha+\beta-1)e^{-t}-(\alpha+1)e^t$$
$$y(t)=x_1+x_2=1+(\alpha+\beta-1)e^{-t}$$

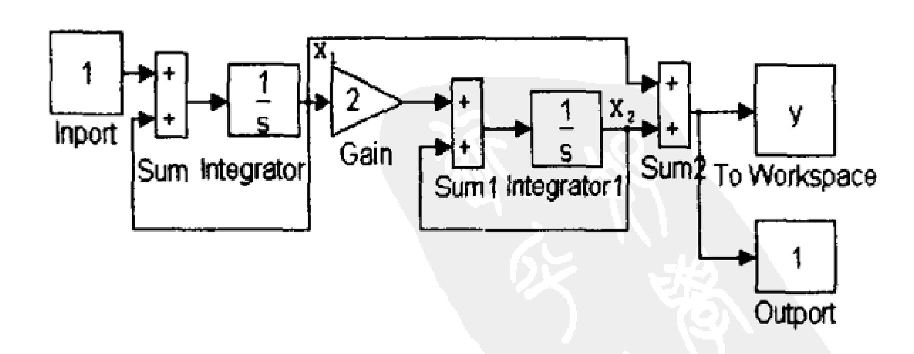


图 3-20 SIMULINK 系统结构框图 $G(s)K(s)$

由其方程解可见，当 $t \to \infty$ 时

$$x_1 \to \infty, \quad x_2 \to -\infty, \quad y \to 1$$

因此，尽管系统的内部变量是发散的，然而系统的输出仍能在任何初始条件下收敛为 1 。如果一个系统中所有可能的传递函数在所有的输入与输出范围内都是稳定的，则我们称这个系统为

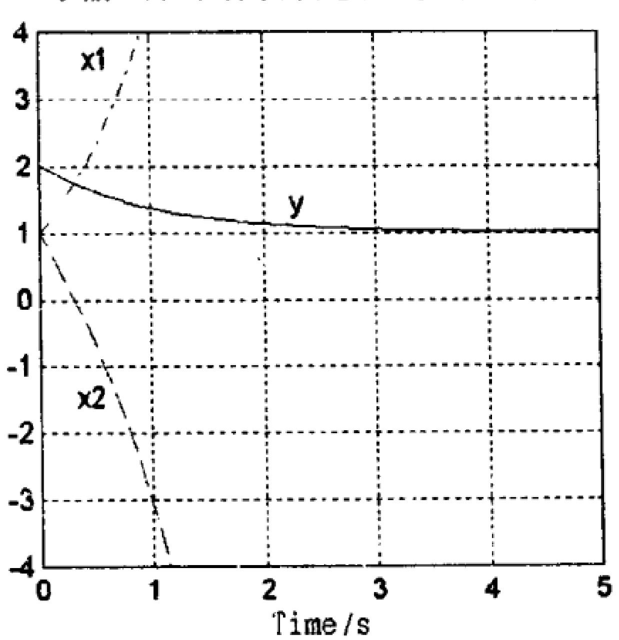


图 3-21　初始条件为(1,1)时的阶跃响应

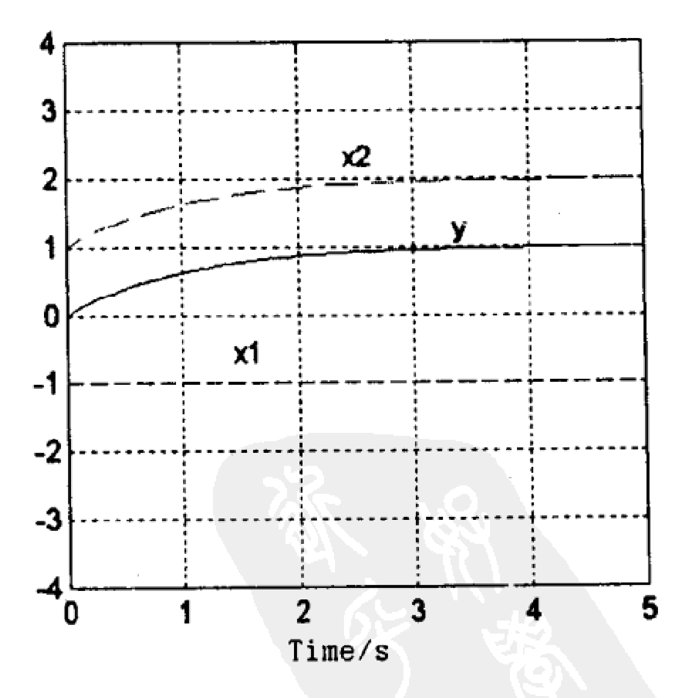


图 3-22　初始条件为(−1,1)时的阶跃响应

固有的稳定系统。因此，上述系统不属该种系统。同时，我们也

注意到，系统有两项 e^t 与 e^{-t}，其中发散项被收敛项抵消。

总之，当一个极点被一零点抵消时，它在系统内部建立一个隐蔽的工作模式，如果这个极点是不稳定的，它会导致这个系统内部不是稳定的，这是不可接受的。对于熟悉状态空间概念的读者，可以说该系统为不能观的（如果这个模式是不稳定的，则为不能测的）。然而，如果初始条件 $\alpha=-1$，则所有的变量都不会发散。除此一点外的任何初始条件，我们都可得到上述结论。对于初始条件为(1,1)与(−1,1)的阶跃响应结果见图 3-21 与图 3-22 。

【例 3-6】其它功能块的例证。

该例介绍其它四种功能块：From Workspace(Sources 库)、DeMux 与 Mux(Conections 库)和 Switch(Nonlinear 库)。该例是简单的， 主要用于验证这些有用的功能块。

From Workspace 功能块允许从 MATLAB 工作空间取得数据作为方块图的输入。数据的第一列是一个升阶矢量（时间轴），其它列为数据矢量。

Mux 功能块以多个矢量为其输入，然后将这些矢量排列为一矩阵。例如，如果想在示波器中同时观察几个矢量，可以使用这个功能块传输这些矢量。 DeMux 执行上述的反向功能；它把一个矩阵输入变成列矢量组。

Switch 是一个条件功能块。它有三个输入端；该方块的第二个输入作为条件测试端。如果第二个输入端大于或等于 0 ，则 Switch 的输出端等于第一个输入端，否则， Switch 的输出端等于第三个输入端。

为验证这些方块功能，将在 MATLAB 空间生成三个输入矢量。第一个输入为正弦波；第三个输出为第一输入的反值；第二个输入为 1 与−1 的交替矢量。因此，其输出值为 sin*t* 与−sin*t* 的交替采样值。结构框图见图 3-23 。在 MATLAB 工作空间，首先建立如下矢量：

```
》 t=[0： pi/50： 2*pi]'； x=sin(t)； y=-sin(t)；
》 ind=(-1).^[0:100]'； % 第二个输入
```

然后进入 SIMULINK 环境，建立结构框图。第一个方块从 MATLAB 工作空间取输入矢量。它的输出是一个具有三列矢量的矩阵。 DeMux 方块把这个矩阵分成 3 个列矢量分别送入 Switch 的输入端。 Switch 的输出、矢量 x 与矢量 y 通过 Mux 功能块汇集在一起送到示波器。启动仿真程序，会得到其绘图结果，见图 3-23 。注意实际上为三个绘图轨迹，矢量 x 与 y 形成了 Switch 输出的包络线。

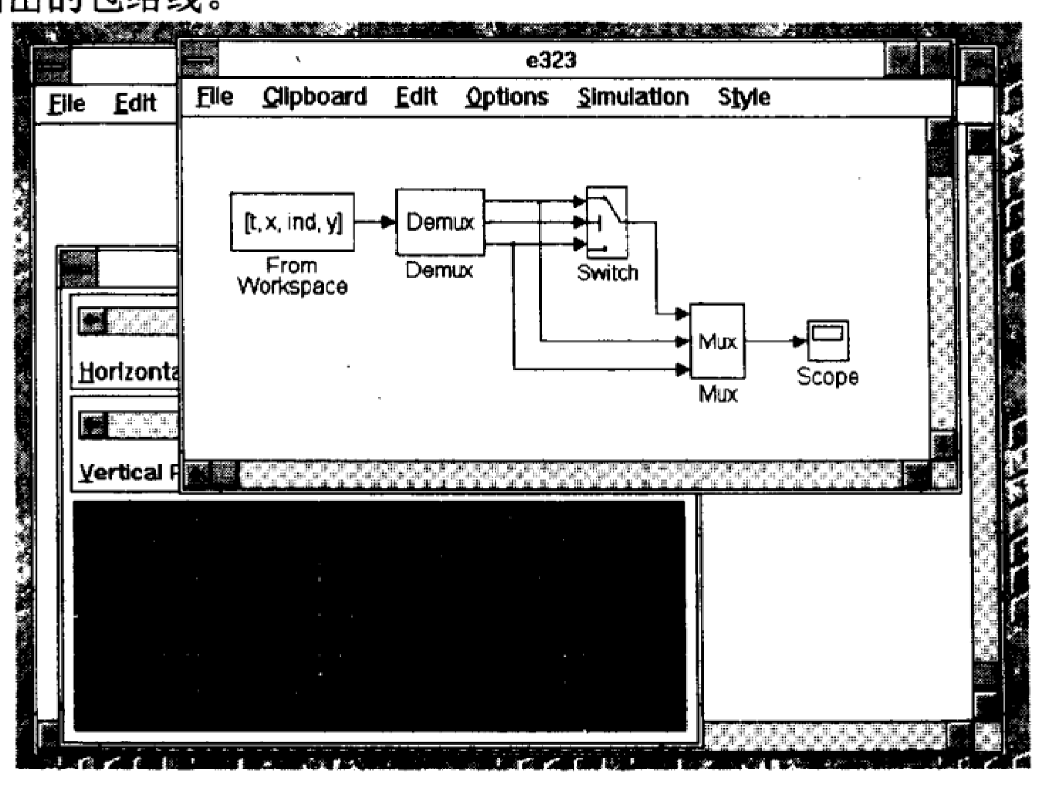


图 3-23　例 3-6 的机构框图及示波器输出波形

3.6.4　SIMULINK 的高级特征

在此还简短地介绍 SIMULINK 的其它几种高级用法，但是不再举例说明。当建立了一个结构框图的时候， SIMULINK 就针对这个模型建立了一个 M 文件（函数）。这些函数被称为 S 函数（ s-functions ）。与其它 MATLAB 函数一样，用于打开这些文件并进行编辑与修改。用于建立 S 函数的 MATLAB 命令是 sfun 。该命令表明可以不用通过图形接口，而用编写的程序去直接仿真一个系统。一个 S 函数，可以在 SIMULINK 环境下转换成一个方块图。同样，也可以通过使用 Option 菜单中的 Mask 命

令给 SIMULINK 创建新的功能块。使用 Mask 选项，可以选取任何存在的功能方块或 S 函数，并且改变它们的特性。在 Extra block 库中存有许多 Mask 功能块。这些 SIMULINK 的 S 函数与 Mask 特征都与 MATLAB 的操作原理相同，可以把它们作为 M 文件打开，修改或添加一些新的功能。有关这方面详细内容，请见 SIMULINK 用户手册。

习　题

3-1　分析改变欠阻尼二阶系统的极点位置对系统的影响，并观察系统的阶跃响应与伯德图之间的关系。欠阻尼的二阶标准系统传递函数如下：

$$G(s)=\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}=\frac{(\omega_a^2+\sigma^2)}{s^2+2\sigma s+(\omega_a^2+\sigma^2)}$$

极点位置：

$$s=-\sigma\pm \mathrm{j}\omega_a$$

式中，$\sigma=\zeta\omega_n$；$\omega_a=\omega\sqrt{1-\zeta^2}$；$\zeta=\cos(\theta)$ 。

初步分析：利用二阶系统的特性与其方程表达式，讨论分析系统每一个参数σ、ω_a、ω_n与ζ变化（其它参数不变）时，对系统的阶跃响应特征参数($\sigma\%,T_r,T_s,T_p$)与伯德图参数(M_r， Bw)的影响。

上机实验：对下述每一部分，阶跃响应对应的时间从 t=0 至 t=10 ，同时伯德图的频率范围ω为 0.1 ～ 10 。对于伯德图中的幅频特性，频率轴与幅值图都是取对数的，且绘制在屏幕的上半部分；相频特性是半对数坐标，绘制在屏幕的下半部分。必须完成总共 12 张图，其中含有 36 个仿真结果（其中有些是重复的，仅有 27 个仿真结果是必要的）。因为要做大量的重复仿真，最好使用 for 循环指令写一简单循环程序，这样可节省许多时间。

(1) 设ω_a=1 ，让σ= 0.5 、 1 、 5 ，求阶跃响应与伯德图。

(2) 重复(1)部分，设σ=1 ，而让ω_a= 0.5 、 1 、 5 。

(3) 重复(1)部分，设$\zeta=1/\sqrt{2}$ ，而让$\omega_n=\sqrt{2}/2$ 、$\sqrt{2}$ 、 $5\sqrt{2}$ 。

(4) 重复(1)部分，设$\omega_n=\sqrt{2}$，而让 θ=30°、 45°、 60°。

注：ζ=cos θ

(5) 由上述得到的数据与图形，计算出所有上述 6 种响应参数，列表

显示计算结果。分析参数变化（增加、减少与不变）对阶跃响应的影响并给出结论。

下面是关于阶跃响应与频率响应的关系的问题。

(6) 在 M_r 与 σ %之间存在着什么关系？能够定量表示它们之间的关系吗？

(7) 在 T_r 与 Bw 之间存在着什么关系？可以定量表示它们之间的关系吗？

(8) 在 ω_n 与 Bw 之间存在着什么关系？可以定量表示它们之间的关系吗？ω_n 可以近似地表示 Bw 吗？

3-2 这一习题主要研究对二阶系统添加零点与极点的作用。零点与极点是分别加入的。添加零点（或极点）时又分为两种情况，在环内或者在环外。由于在经典系统设计中，经常使用增加零、极点（超前-滞后补偿器）的方法来修正系统的动态特性，因此掌握它们对系统的调节作用是十分重要的。

最小相位系统 $G(s)$ 构成单位反馈闭环系统 $T(s)$。上述两种增加零、极点方法的系统结构框图见图 3-24 。

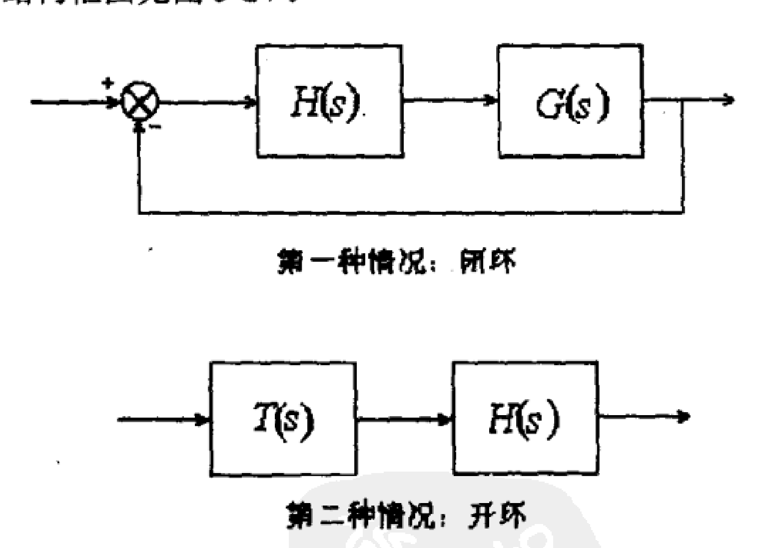


图 3-24　两种增加零、极点方法的系统结构框图

使用粗略的手工根轨迹法、伯德图方法或其它简易分析法，初步分析在上述两种情况下增加零、极点的作用，对响应速度、超调、频宽和系统的稳定性（增益裕量与相角裕量）加以评价。

上机实验：

设 $H(s)=1/(ps+1)$　为增加极点环节。

$H(s)=zs+1$　　为增加零点环节。

注意对应的实际极点与零点为$-1/p$ 与$-1/z$ 。与习题 3-1 相同，可绘制一些图并进行数据分析。由于仿真的数据较多，应该选取一些容易记忆的变量名，并记录下来作为参考。也由于生成的矢量数较多，有可能超出你的计算机的存储空间。为减小计算机和占用空间，可以把实验分为两步。例如，首先完成增加极点的仿真，然后存储仿真的结果与数据，并清除工作空间。第二步再完成增加零点的仿真，再以不同的文件名存储。所有的阶跃响应仿真时间 t 范围都是从 0 至 10 。伯德图中的频率ω 轴的定标范围是ω=0.05~20 。因为仿真的数量多,编制一个简单的程序是必要的。p 与 z 都取下列数值:

{0. 2624， 0.6122， 1.4286， 3.3333}。

这些数值可以由下列 MATLAB 命令生成:

```
》 b=log(7/3)； a=(10/3)exp(-4*b)；
》 for i=1:4, p=a*exp(b*i),end ;
```

对于每一种情况，必须求得下列的仿真结果与其对应的数据，并将其数据列表。

第一种情况：需要做出闭环阶跃响应($\sigma\%$， T_r 与 T_s)，开环伯德图(ω_{gc} 与 PM)，闭环伯德图(M_r 与 Bw)。

第二种情况：需要做出阶跃响应与伯德图（由于这种情况没有闭环，因此只有一种伯德图）。

(1) 求 $T(s)$的阶跃响应与伯德图和 $G(s)$的伯德图。注意这是一个最小相位系统。

(2) 对于第一种情况，增加一个极点，让 p 取特定值，得到适当的响应。 绘制与(1)项相应的响应，并进行比较。例如，阶跃响应图中应有五条轨迹，其中两条应为(1)项。

(3) 对于第二种情况，给 $T(s)$添加一个极点。完成如(2)项的分析并给出结论。

(4) 对于第一种情况，给 $G(s)$添加一个零点，完成如(2)项的分析并给

出结论。

(5) 对于第二种情况，给 $T(s)$添加一个零点，完成如(2)项的分析并给出结论。

注意到给传递函数 $T(s)$增加一个极点或一个零点，并不改变它的稳定性，然而这样做却改变了它的其它特性。若给开环传递函数 $G(s)$加入一个极点或零点，既改变了它的稳定性，也改变了它的响应参数。一般地讲，加入一个零点，会提高系统的稳定性；加入一个极点，可能会引起系统不稳定。这就是为什么补偿器加在环内的原因之一。这些零极点的位置会在很大程度上决定系统的特性参数。

(6) 基于上述结果，试分析给 $G(s)$中加入一个极点后对σ %、T_r、PM 和 Bw 的影响是什么？

(7) 试分析给 $T(s)$加入一个极点后，对σ %、T_r、PM 和 Bw 的参数影响是什么？

(8) 试分析给 $G(s)$加入一个零点后,对σ%、T_r、PM 和 Bw 的参数影响是什么？

(9) 试分析给 $T(s)$加入一个零点后，对σ %、T_r、PM 和 Bw 的参数影响是什么？

下面是关于阶跃响应与频率响应参数之间对应关系的问题。

(10) 在σ%与 PM 之间存在着何种关系？

(11) 在ω_{gc}与 Bw 之间存在着何种关系？

3-3 该习题是研究非最小相位零点的作用（即零点位于右半平面）。带有右零点或时间延迟环节的系统称为非最小相位系统。系统传递函数为

$$G(s)=\frac{n(s)}{s^2+0.5s+1.5}$$

(1) 设 $n(s)=1.5$，利用二阶系统阶跃响应的计算公式估算这个系统的超调量、峰值时间与过渡过程时间，使用计算机求得系统的阶跃响应，并比较所得到的结果。

(2) 设 $n(s)=(-s+\alpha)/\alpha$,求得 $\alpha=\{1,3,6\}$时的系统阶跃响应。

(3) 设 $n(s)=(s+\alpha)/\alpha$,求得 $\alpha=\{1,3,6\}$时的系统阶跃响应。

分别绘制两个阶跃响应图，并列表表示所得到的结果。(2)与(3)项中的

阶跃响应中应附有(1)项情况下的阶跃响应曲线，以作为比较。

(4) 基于所得到的数据，试述在阶跃响应中左平面零点与右平面零点各自的作用是什么？从实验中可以注意到，当系统具有右平面零点时，阶跃响应曲线首先朝期望值相反方向变化。当系统具有奇数个右零点时，就会出现这种现象。

3-4 本习题介绍了滞后与超前补偿器在不稳定系统中的补偿作用。以双积分器系统为例

$$G(s)=1/s^2$$

其补偿器有如下三种。

Ⅰ. 比例补偿器：$K(s)=K_c$

Ⅱ. 超前补偿器：$K(s)=K_c(s+1)/(s+5)$

Ⅲ. 滞后补偿器：$K(s)=K_c(s+5)/(s+1)$

初步分析：利用手工绘制系统根轨迹与伯德图，并回答下列问题。

(1) 在第Ⅰ种情况下，系统能够稳定吗？增加 K_c 的作用是什么？

(2) 在第Ⅱ种情况下，系统能够稳定吗？增加 K_c 的作用是什么？

(3) 在第Ⅲ种情况下，系统能够稳定吗？增加 K_c 的作用是什么？

上机实验：

(4) 设 $K_c=\{0.1,0.5,1\}$，对上述每一种情况进行仿真，求得其阶跃响应。要求将这三种情况的闭环阶跃响应绘制在一张图中。分析该图，证明在初步分析过程中所得到的结论。在每种情况中，仿真时间可为{10， 50， 5}。

3-5 本习题是关于验证闭环系统的抗扰动特性并与开环系统进行比较。首先分析并回答问题，然后通过仿真验证所得的结论。

初步分析：以一阶系统 $G(s)$为例， $G(s)=1/(s+1)$

(1) 开环控制，配置一补偿器 $K(s)$，它的极点位于$-1 \pm j$，并使系统的阶跃响应与斜坡响应的稳态误差为零。

(2) 反馈控制，设系统为补偿器与被控对象串联的单位反馈结构。求一补偿器，该补偿器加入之后应使系统满足(1)项的要求。

(3) 假设在 $K(s)$与 $G(s)$之间存在一个单位阶跃扰动，求上述两种系统对此扰动信号的阶跃响应，问哪一种对此扰动信号抑制作用强？

(4) 通过观察上述两种情况的伯德图来评价它们的滤波器特性。哪一

种系统能够对高频噪声或干扰具有较好的抑制作用？

上机实验：

(5) 求上述每一种情况的阶跃响应来验证(3)项的结果。

(6) 设输入信号为 $n(t)=0.01*\mathrm{rand}(t)$，使用 lsim 命令仿真这个噪声信号响应。在每一种情况下，把噪声信号与其响应绘制在同一图中。仿真结果与分析结果一致吗？在抑制低频干扰信号与抑制高频干扰信号之间是否存在着冲突？

3-6 本题介绍比例-积分(PI)控制,并研究它的作用。考虑一典型的二阶系统 $G(s)$与一个串联 PI 补偿器 $K(s)$:

$$G(s)=\frac{\omega_n^2}{s(s+2\zeta\omega_n)} \quad K(s)=K_p+\frac{K_I}{s}=K_c\frac{s+z}{s}$$

初步分析：

(1) 选择 K_c 与 z 值，使闭环系统稳定。

(2) PI 控制对系统的稳态误差特性的影响如何？

(3) 设$\zeta=0.5,\omega_n=2,K_c=1$ 和 $z=1$ 。绘制补偿前与补偿后的系统根轨迹图；试评述 PI 控制对系统的上升时间与超调量$\sigma\%$的影响。

(4) 利用开环与闭环伯德图，试评述 PI 控制对系统的谐振峰值、闭环频率带宽及相位裕量的影响。

上机实验：

(5) 利用根轨迹法求满足(3)项特定参数的闭环极点位置。

(6) 求在补偿前与补偿后的系统阶跃响应。比较它们的上升时间与超调量$\sigma\%$。

(7) 求在补偿前与补偿后的系统伯德图。比较它们的相位裕量与增益穿越频率。

(8) 求在补偿前与补偿后的系统闭环伯德图。比较它们的频带宽度与谐振峰值。

注：将补偿前与补偿后的结果绘制在同一图中，以利于比较。

3-7 本习题是 PID 控制系统的设计问题，考虑如下一个不稳定系统。

$$G(s)=\frac{1}{(s-1)^2}$$

设计要求：闭环稳定，且单位阶跃响应稳态误差为 0 。

初步分析：

(1) 利用根轨迹法分析可知，使用串联 PID 补偿器 $K(s)$ 可以达到上述两个方面的要求。注意在考虑系统稳态误差之前，系统必须是稳定的。设系统为单位反馈系统结构。 PID 补偿器传递函数为

$$K(s) = K_{\mathrm{p}} + \frac{K_{\mathrm{I}}}{s} + K_{\mathrm{D}}s$$

上机实验：

(2) 使用试探方法选择 PID 参数，满足上述特性要求。然后求其系统的阶跃响应，并绘制其响应图形。注意到 PID 补偿器引入了两个零点。这些零点可以为复数或实数。请分别对两种情况进行实验。

(3) 假定所有的闭环极点都在 $s=-1$ 处，选择其补偿器参数，完成满足特性指标要求的系统补偿器设计。绘制对应这种情况下的根轨迹图及阶跃响应图。

3-8 本习题比较两种补偿方法（串联法与反馈法）的作用。被控对象 $G(s)$与 PI 补偿器 $K(s)$传递函数如下：

$$G(s) = \frac{4}{s(s+2)} \qquad K(s) = \frac{s+1}{s}$$

初步分析：

(1) 说明 $K(s)$与 $G(s)$串联组成的单位反馈系统是稳定的，并且其阶跃稳态误差与斜坡稳态误差都为 0 。

(2) 试说明如果 $K(s)$放置在反馈通道，尽管两个系统具有同样的闭环极点，而该系统将不具有上述性质。

(3) 说明如果在闭环的外面再放置一个补偿器，就可以达到(1)项所得到的特性。请试写出这个补偿器的传递函数 $K_{\mathrm{f}}(s)$。

(4) 假设由于元件的误差或其它原因，其补偿器参数改变为

$$K_2(s) = \frac{s+0.5}{s}$$

试重复(1)项工作，说明系统是否还具有原来补偿后的性质。

(5) 使用 $K_2(s)$作为补偿器，重复(2)项的工作。

(6) 使用(3)项中的 $K_{\mathrm{f}}(s)$，说明(3)项中的跟踪特性是否还存在。

上机实验：

(7)绘制上述每一种情况的阶跃响应与斜坡响应图形。求每一种情况下的系统阶跃响应误差与斜坡响应误差。并与初步分析结果进行比较。

3-9 本题说明当使用伯德图对非最小相位系统进行分析时，可以导致错误的结论。所讨论的控制对象传递函数为

I $KG_1(s)=\dfrac{10K(-1+s)}{s(1+10s)}$ $K=-0.05$

II $KG_2(s)=\dfrac{K(s-1)}{s(2s+1)}$ $K=-0.5,\ -1,\ 1$

III $KG_3(s)=\dfrac{K(4s-1)}{(10s-1)(2s+1)(s+1)}$ $K=-1.5,\ -1,\ 1$

IV $KG_4(s)=\dfrac{7K}{(s-1)(s+2)(s+4)}$ $K=2$

初步分析：

(1) 使用根轨迹法求 K 值的稳定范围。对应于这个稳定范围求稳定裕量的上下限。

(2) 绘制奈魁斯特图与使用奈魁斯特稳定判据验证(1)项中答案。此外，求其相角裕量。

上机实验：

(3) 对应特定增益绘制其伯德图，并且使用 margin 命令求其增益与相角裕量。比较(1)项与(2)项的结果。试给出使用伯德图决定非最小相位系统伯德图的有效性。

3-10 被控对象的开环传递函数如下：

I $G_1(s)=\dfrac{0.485}{s^2+0.3s+1}$

II $G_2(s)=\dfrac{7(s+2)}{s(s^2+2s+12)}$

(1) 利用开环伯德图求系统的增益裕量与相角裕量。

(2) 利用闭环伯德图求系统的谐振峰值。

(3) 求系统的闭环极点（单位反馈），并且计算系统阻尼率。

(4) 求系统闭环阶跃响应，求$\sigma\%, T_r$与 T_s，并对系统的瞬态响应给予评价。

3-11　本题研究两个非线性系统的特征，显示它们的不规则特性。在给定的参数与初始条件下进行系统仿真。绘制对应时间变化的状态变量图。可以改变参数及初始条件，大范围地观察系统特性的不同变化。

(1) Rossler 系统

$$\dot{x}=-y-z \qquad x(0)=5$$

$$\dot{y}=x+\frac{y}{5} \qquad y(0)=5$$

$$\dot{z}=\frac{1}{5}+zx-5.7z \qquad z(0)=5$$

(2) Duffing 系统

$$\dot{x}=y \qquad x(0)=5$$

$$\dot{y}=-(x^3+x+y)+\cos\theta \qquad y(0)=5$$

$$\dot{\theta}=\omega \qquad \theta(0)=3$$

3-12　本习题说明在非线性系统中计算误差的影响。使用不同的积分算法重做 3-11 题中的仿真过程，绘制响应过程曲线，比较采用不同算法得到的状态变量过程曲线的差异。由此看到系统对计算误差是很敏感的。

3-13　本习题将证明具有相同的增益或相角裕量的系统，可以有非常不一样的阶跃响应。因此，系统虽然具有较好的裕量，但并不表示系统具有很好的阶跃响应。

(1) 对于下面每一个给定的传递函数，绘制它们的奈魁斯特图或伯德图，计算与比较它们的增益裕量或相角裕量。

$$G_1(s)=\frac{1.4(1+0.5s)\mathrm{e}^{-0.4s}}{s^2} \qquad G_2(s)=\frac{0.072\mathrm{e}^{5s}}{s(1+5s)^2}$$

$$G_3(s)=\frac{1.65\mathrm{e}^{-12s}}{1+20s} \qquad G_4(s)=\frac{1.25\mathrm{e}^{-15s}}{(1+5s)^2}$$

$$G_5(s)=\frac{0.1}{s(1+5s)^2} \qquad G_6(s)=\frac{1.4(1+1.2s)\mathrm{e}^{-0.2s}}{s^2}$$

(2) 设系统为单位反馈系统。求闭环阶跃响应，并计算$\sigma\%$与T_s，同时绘制闭环系统伯德幅频图，且计算其谐振峰值M_r。

(3) 将上述结果列表，根据表中结果试分析稳定裕量与时间响应之间的关系，并给出结论。你认为哪一个频率响应参数(GM、PM、M_r)是最能可靠地表示阶跃响应中$\sigma\%$的品质。

3-14 一个标准机械振动是一个二维自由度问题。有两块物体：第一块物体通过弹簧和阻尼器与一固定墙面相连，在两块物体中间也有弹簧和阻尼器相连，见图 3-25 。一个按正弦规律变化的力作用在第二块物体上，求这两块物体的位置与速度。该运动方程的参数如下：

$$M_1\ddot{x}_1 + (K_1 + K_2)x_1 + (c_1 + c_2)\dot{x}_1 - K_2x_2 - c_2\dot{x}_2 = 0$$

$$M_2\ddot{x}_2 + (K_2x_2 + c_2\dot{x}_2 - c_2\dot{x}_1 - K_2x_1 = F\sin(20t)$$

$$M_1 = 10\text{kg},\ M_2 = 20\text{kg},\ K_1 = 1000\text{N/m},\ K_2 = 200\text{N/m},$$

$$c_1 = c_2 = 120\text{N}\cdot\text{s/m},\ F = 100$$

建立上述系统模型，对系统进行仿真（时间为 4s ）。绘制四个状态（两个位置与两个速度）对应时间变化的曲线。将它们对应地绘制在同一图中，x_1 与 x_2 、$\dot{x}_l$ 与 x_1 和 $\dot{x}_2$ 与 x_2 对应。

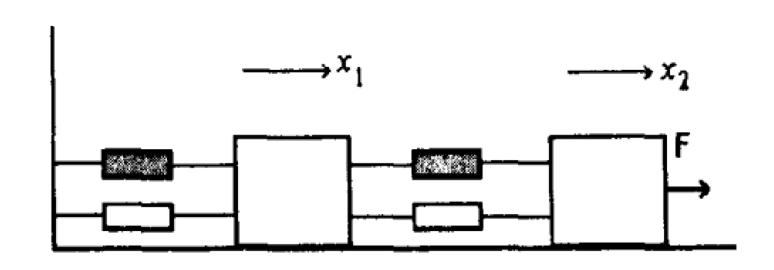


图 3-25 两维自由度问题

第 4 章　经典控制系统设计方法

4.1 引言

经典控制系统经常使用传递函数来描述。最常用的两种设计方法是根轨迹法与频率法。常用的闭环系统特性参数有稳态误差与阶跃响应参数，如上升时间 T_r、过渡过程时间 T_s、超调量σ%等。有关这方面的定义，请参见有关系统设计方面的书。在假定系统具有一对闭环主导极点的条件下，系统的稳定裕量（增益裕量与相角裕量）与系统的阶跃响应参数之间存在着一定的关系。而在主导极点附近的零点与附近的极点将会影响系统的阶跃响应。这些内容在有关的专业书籍中都有详细的讨论。

采用计算机对控制系统进行辅助设计的优点之一是可以对系统的整体模型进行描述与分析。本章将初步介绍控制系统计算机辅助设计方法，主要内容有闭环系统的时域与频域分析方法和依照系统性能指标的要求进行系统补偿的方法。在讨论这些特殊的设计方法之前，首先复习根轨迹与伯德图法是如何描述闭环系统特性的。

4.1.1 根轨迹法

根轨迹是描述当一特殊传递函数参数(通常为增益)变化时，闭环系统极点位置的变化轨迹。根轨迹是确定闭环系统的绝对稳定性与相对稳定性的非常好的方法。即使采用伯德图设计，根轨迹法也可以用于系统的稳定性分析。对于一个二阶系统，在 s 平面中可以得到过渡过程时间 T_S、阻尼频率 ω_d、阻尼系数 ζ 和自然频率 ω_n(对应于欠阻尼情况)的恒值轮廓线。这样会给设计带来很大的方便。二阶特征方程根的表达式如下：

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -\delta \pm j\omega_d$$

式中，$\delta = \zeta\omega_n$；$\omega_d = \omega_n\sqrt{1-\zeta^2}$ 。

标准的零初始状态阶跃响应的数学表达式如下：

$$y(t) = 1 - \frac{1}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \cos(\omega_d t - \arcsin\zeta)$$

过渡过程时间 T_s 常取为 4 倍至 5 倍的时间常数，因此

$$4\tau < T_s < 5\tau$$

式中，$\tau = 1/\lambda$，$\lambda = \zeta\omega_n$

尽管对于上升时间 T_r 与延迟时间 T_D 不存在准确的关系表达式，但是他们都与 ω_n 成反比。超调量是用百分数形式给出的。

$$\sigma\% = 100e^{\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}}\% \qquad (0 \leqslant \zeta < 1)$$

例如：设超调量<25%，过渡过程时间<5s ，则从以上参数，可以确定

$$\zeta > 0.4 \rightarrow \theta < 66^\circ$$

$$T_s < 5 \rightarrow \lambda < -1$$

对于给定的技术性能要求，对应在 s 平面上的区域，见图 4-1 中的阴影区。

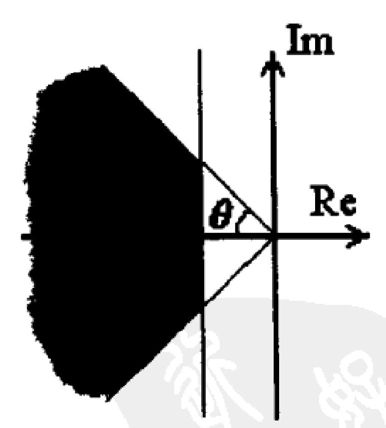


图 4-1　可允许的极点位置区域

上述假设成立的前提是系统必须具有一对共轭复数主导极

点，即该极点对系统特性起主导作用。否则，如果实际系统具有靠近这些共轭复数主导极点的零点与极点，按照此法预测的系统特性将是不准确的。

4.1.2 伯德图法

伯德图法可以用于确定闭环系统的稳态误差、阻尼系数、自然频率（对应主导极点）与稳定裕量。对于非最小相位系统，建议不要使用伯德图法去判定系统的稳定性，而可以采用奈魁斯特稳定判据。

假设在单位反馈系统中，被控对象的传递函数如下：

$$KG(s)=\frac{K\omega_{\rm n}^2}{s^2+2\zeta\omega_{\rm n}s+\omega_{\rm n}^2}$$

从开环系统的伯德图中，可以确定系统是哪一类系统。在该种情况下，初始斜率为零。因此，$KG(0)$是有限的数。这就意味着系统是 0 型系统。$K_{\rm P}=KG(0)=K$ （K 为低频增益)。系统具有$-20n$dB/dec 的初始斜率的系统为 n 型系统。

该系统的闭环传递函数为

$$T(s)=\frac{K\omega_{\rm n}^2}{s^2+2\zeta\omega_{\rm n}s+(K+1)\omega_{\rm n}^2}=\frac{K\omega_{\rm n}^2}{s^2+2\zeta_{\rm cl}\omega_{\rm ncl}s+\omega_{\rm ncl}^2}$$

式中，$\omega_{\rm ncl}=\omega_{\rm n}\sqrt{K+1}$；$\zeta_{\rm cl}=\zeta/\sqrt{K+1}$

对于大 K 值的情况，增益穿越频率$\omega_{\rm gc}$（$\omega_{\rm gc}\approx\omega_{\rm n}\sqrt{K}$）给出了闭环自然频率的精确测量值。闭环系统的频带宽也可以由$\omega_{\rm gc}$ 近似表示。然而，实际闭环系统的频带宽度可变化到该值的 100%。

闭环系统的阻尼大约为开环系统阻尼系数的 $1/\sqrt{K}$。因此闭环系统比开环系统具有较快的动作，同时也具有较大的超调量。那么闭环系统的阻尼系数$\zeta_{\rm cl}$ 能否在开环系统伯德图中求得呢？由奈魁斯特图中相角裕量 PM 的定义，并利用简单的几何方法，可以得到

$$\mathrm{PM}=2\arcsin\frac{1}{2|T(\mathrm{j}\omega_{\rm gc})|}$$

因为$\omega_{gc}\approx\omega_n\sqrt{K}$，那么

$$\mathrm{PM}\approx 2\arcsin(\zeta_{cl})$$

或

$$\zeta_{cl}\approx\sin(\mathrm{PM}/2)$$

也可以采用经验公式来估算ζ_{cl}（°）

$$\zeta_{cl}\approx 0.01\mathrm{PM}$$

由相角裕量和增益裕量可以确定系统的相对稳定性。增益裕量 GM 的定义如下：

$$\mathrm{GM}=-|G(\mathrm{j}\omega_{pc})|_{dB}$$

式中，$\angle G(\mathrm{j}\omega_{pc})=-180°$

根轨迹法或伯德图法都可用于分析闭环系统特性。根轨迹法可以给出闭环极点图形，然而，伯德图法可以用于确定闭环系统的稳态误差、频带宽与稳定裕量。

4.2　系统补偿

控制系统补偿有各种方法。其中有串联(caseade)补偿法、反馈补偿法和串联-反馈补偿法。在串联补偿法中，补偿器与被控制对象串联。在反馈补偿法中，补偿器被置于反馈通道。在这两种情况中，系统的开环传递函数与闭环传递函数是一样的。因此，它们具有相同的根轨迹图与伯德图，且具有同样的稳定性。然而，它们的闭环零点是不相同的，因此具有不同的稳态误差。由于反馈会减小在前向通道参数变化的影响，因此串联结构具有更好的敏感特性。在单位反馈情况下，也容易控制误差的恒定。正因为如此，串联结构的补偿方法用得更多。

在闭环的外面放置一个滤波器，可以过滤外部传送来的信号。对特殊已知频率进行限制的陷波滤波器为常用的滤波器。由于这些滤波器被放在闭环外面，因此它们不会受反馈环节的调节。所以，只有在事先准确知道要消减的频率时，才使用这种滤波方法。

在课堂上学到的最普通的补偿方法是零极点对消的方法。采用一个位置相同的补偿器零点（极点），对消掉被控制对象中

的主导极点（零点），然后再把所希望的系统极点添加到补偿器的分母中。

若要抵消的零极点出现在左半平面，可采用上述方法。但是，对于出现在右半平面的零极点，不宜采用零极点对消法。存在的问题之一是不可能知道极点和零点的真正位置。所有的被控对象传递函数仅是一个近似的数学模型。它不可能完全准确。零极点对消的另外一个问题是，即使对消的零极点在系统的输入-输出传递函数中消失，然而，它们有可能在其它形式的传递函数（如扰动对输出的传递函数）中再次出现。因此，可能有好的输入跟随特性，但是却具有差的抗扰特性。如若采用零点对消方法，设计者一定要检验当改变零极点位置时，闭环系统响应的敏感程度。

实际中，普遍采用的控制器有两种：PID 控制器与超前-滞后补偿器。这些控制器的传递函数为

PID $\qquad K_P + K_D s + \dfrac{K_I}{s}$

超前-滞后 $\qquad K\dfrac{s+a}{s+b}$

4.3 比例、积分与微分(PID)控制

PID 控制器可以采用几种形式：比例(P)控制器,$K_D = K_I = 0$；比例微分(PD)控制器，$K_I = 0$；比例积分(PI)控制器 $K_D=0$；全 PID 控制器。比例控制是最简单的控制结构。然而，它也能使系统满足某一方面的特性要求，如 GM 、 PM 、稳态误差等。在闭环控制中，加入微分控制，增加阻尼作用，而加入积分控制，则增加了系统的稳态误差精度(系统增型)。PID 控制器可以用于补偿系统达到大多数特性参数的要求。它是目前在过程控制中用得最普遍的控制器，它也被广泛地用于各类控制(模拟，数字和适应控制)中。

4.3.1 Ziegler-Nichols 方法

由于在设计 PID 控制器中，要调整 3 个参数，根轨迹与伯德图设计方法通常不被直接采用。Ziegler 与 Nichols 发展了 PID 调节器设计方法。该方法基于简单的稳定性分析方法。首先，置 $K_D=K_I=0$，然后增加比例系数直至系统开始振荡(即闭环系统极点在 $j\omega$ 轴上)。再将该比例系数乘 0.6，而其它参数按下式计算得

$$K_P = 0.6K_m \quad K_D = \frac{K_P\pi}{4\omega_m} \quad K_I = \frac{K_P\omega_m}{\pi}$$

式中,K_m 为系统开始振荡时的 K 值；ω_m 为振荡频率。然而，该设计方法在设计过程中没有考虑任何特性要求。但是 Ziegler 与 Nichols 发现这种设计方法给予过程控制器提供了好的工作性能。工程师们的多年实践经验证明，这种设计方法的确是一种好的方法。

利用根轨迹法或伯德图方法，可以确定 K_m 和 ω_m。例如，对于给定的被控对象的传递函数，可以得到一根轨迹图。对应穿越 $j\omega$ 轴时的增益即为 K_m，而此点的 ω 值即为 ω_m.另外,对于给定的被控对象传递函数可绘制伯德图。在频率为 ω_{pc} 处确定增益裕量，K_m=10(GM/20),而 $\omega_m=\omega_{pc}$，请注意伯德图只能给出近似的结果。

【例 4-1】设计 PID 控制的 Ziegler-Nichols 方法。

被控对象的传递函数为

$$G(s) = \frac{400}{s(s^2+30s+200)}$$

上述传递函数将用于所有后续的例题(对应 PID 、 PD 和超前设计)中。它具有下例特点：

闭环极点 $=\{-4.2\pm j0.93，-21.59\}$

增益穿越频率= 1.95 rad/s

GM=23dB

PM=73°

该闭环阶跃响应没有超调，而其斜坡输入稳态误差为 0.5 。对应该系统的根轨迹见图 4-2 。使用用 rlocus 及 rlocfind 命令可

以求得穿越增益 K_m=14 和穿越频率ω_m=14rad/s ，使用 Ziegler-Nichols 方程可求出下例增益参数:

$$K_P = 9 \quad K_D = 0.5 \quad K_I = 40$$

图 4-2 未补偿的系统根轨迹图

图 4-3 中，绘制了系统补偿前后的开环伯德图与闭环阶跃响应。被补偿的阶跃响应具有 60%超调量，过渡过程时间为 1.2s, 增益穿越频率为 11 rad/s ，相角裕量为 25°,请见本章附录中程序 1(Ziegler-Nichols 程序)，可以用它求得补偿器参数。

4.3.2 解析方法

如前所述，采用 Ziegler-nichols 方法不能设计一个 PID 控制器去满足一特定闭环系统的特性要求。解析法则可以根据给定的稳态误差和操作特性参数来确定 PID 的参数。PID 控制系统的开环增益为

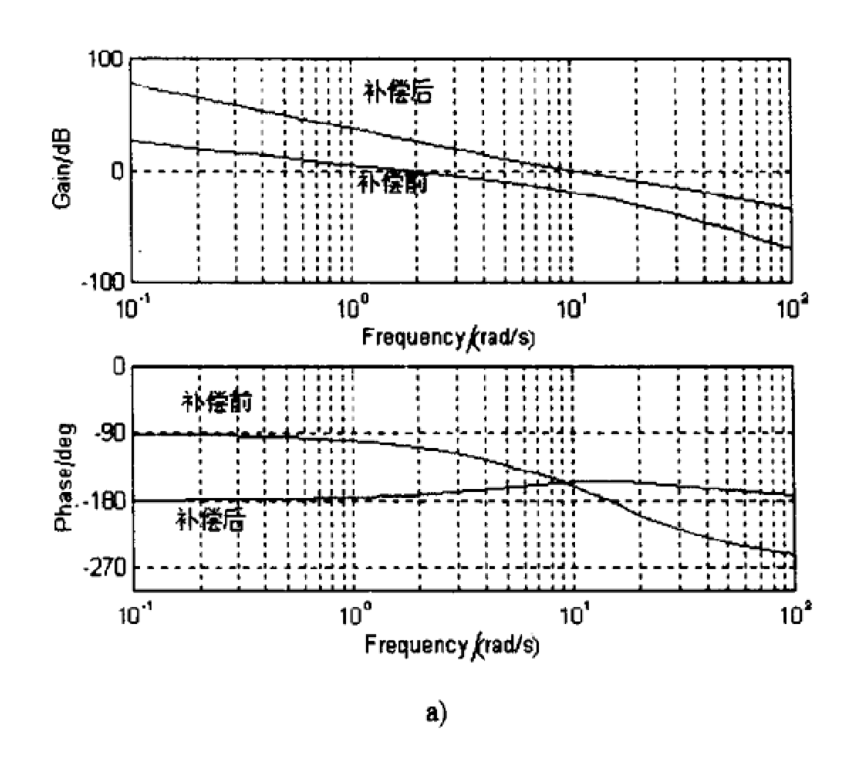
$$(K_P + K_D s + \frac{K_I}{s})G(s)$$

如果 $G(s)$是 n 型系统，补偿后的系统则为 n+1 型系统。误差常数 K_{n+1} 等于稳态误差 e_{ss} 的倒数。

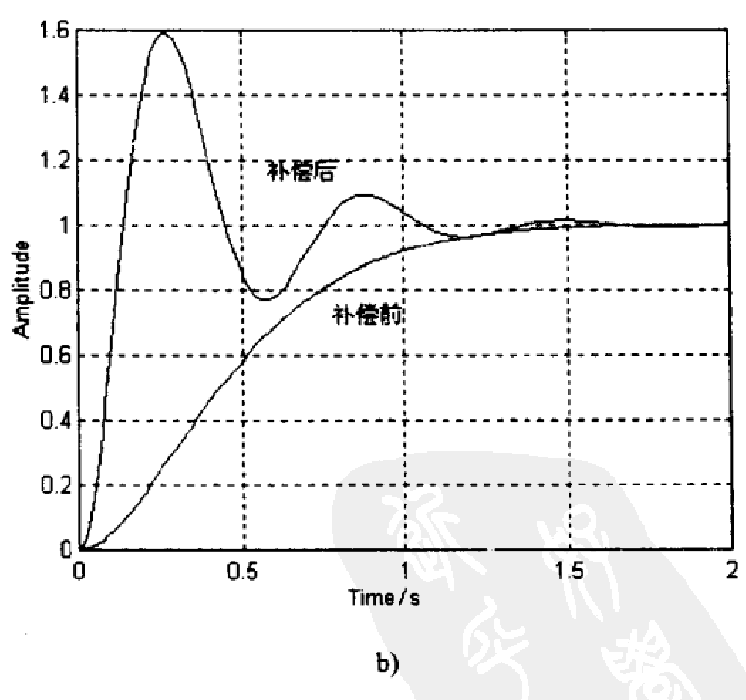
$$K_{n+1} = s^n K_I G(s)\big|_{s=0} = \frac{1}{e_{ss}}$$

对于给定的稳态误差指标，由上面等式可以求得 K_I 的值。由时域指标，如超调量和过渡过程时间，可以确定闭环阻尼系数和自然振荡频率。我们已经知道闭环自然振荡频率对应开环增益

穿越频率ω_{gc}而希望的相角裕量 PM 可以由闭环阻尼系数求出。因此，在$\omega=\omega_{gc}$处，补偿的系统增益应为 1，相角$\theta(\omega_{gc})=-180°+PM$。由上述分析结果(且$K_I$已知)可以写出



a)



b)

图 4-3 开环伯德图与闭环阶跃响应

a）频率响应 b）阶跃响应

$$(K_P + j\omega_{gc}K_D + \frac{K_I}{j\omega_{gc}})G(j\omega_{gc}) = 1e^{j\theta(\omega_{gc})}$$

又可以导出

$$K_P + j\omega_{gc}K_D = \frac{1e^{j\theta(\omega_{gc})}}{G(j\omega_{gc})} + \frac{jK_I}{\omega_{gc}} = R + jX$$

由此可以看出，$K_P = R$，$K_D = X / \omega_{gc}$。

上述过程可使用MATLAB语言程序求解,见本章附录中程序2(解析的PID/PD程序)。

【**例 4-2**】设计PID控制的解析方法

如例4-1被控传递函数为

$$G(s) = \frac{400}{s(s^2 + 30s + 200)}$$

系统的技术指标要求如下：

单位斜坡输入稳态误差 = 0.1

超调量 = 10%

过渡过程时间 = 2s

因为原系统为I型系统，可以求得加入PID控制器后的稳态误差常数。

$$K_2 = sK_IG(s)|_{s=0} = 2K_I = \frac{1}{0.1} = 10 \rightarrow K_I = 5$$

由超调量与过渡过程两项技术指标，可以确定希望的闭环阻尼系数与自然振荡频率 ζ=0.6，ω_n=4 rad/s。因此,给定ω_{gc}= 4 rad/s，PM = arcsin0.6 = 80°,

利用本章附录中程序2,可以得到

$$K_P = 2.02,\quad K_D = 0.52$$

由图4-4,可以看出，正如所期望的系统相角裕量是80°，而增益穿越频率是4 rad/s。闭环系统的阶跃响应见图4-5。由图可见，阶跃响应的过渡过程时间大约为2s。然而。超调量大于技术指标要求，为22%。这是由于PID控制器引入的零点所造成的。

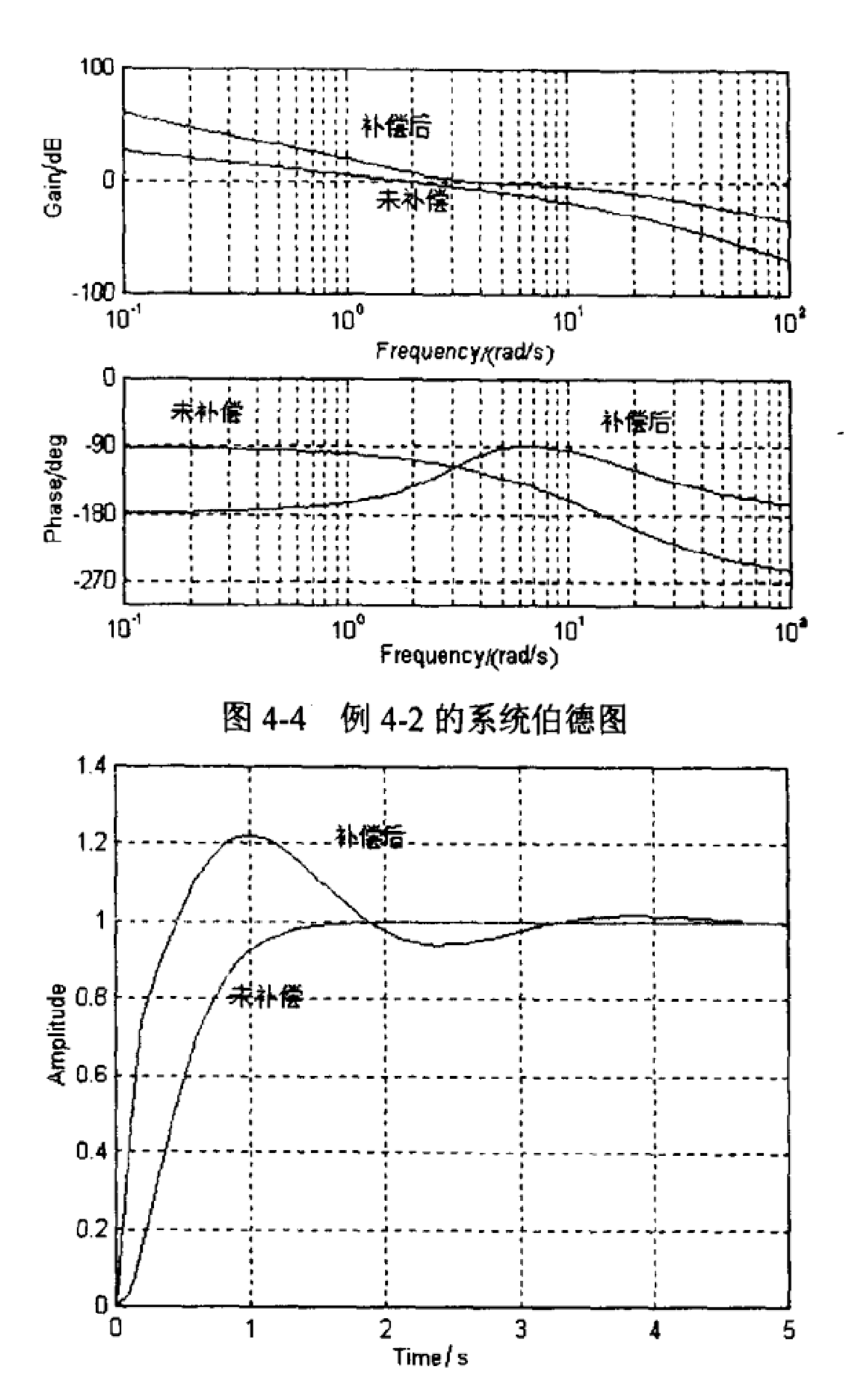


图 4-4　例 4-2 的系统伯德图

图 4-5　例 4-2 的系统阶跃响应

4.3.3　PD 控制

由于 PD 控制应用非常普遍，所以在本章中专门讨论 PD 控制器的设计问题。积分控制多用于系统增型，以减小系统稳态误差。积分控制的弱点是系统增加了极点，使得系统向不稳定的方面变化。相反地，系统增加微分环节，相当于系统增加了零点，使得系统向稳定方向变化。因此，微分控制器常添加在反馈通道

中，特别是在希望增加阻尼比的情况下。有关根轨迹与伯德图的分析方法请见下一节。本节讨论解析的 PD 控制器设计方法。

【例 4-3】 PD 控制。

被控对象的传递函数如前两个例子所示。试设计 PD 控制器，以满足下例特性要求:

$$PM = 45° \quad \omega_{gc}=13.5\ rad/s$$

因为 PD 控制器仅有两个变化参数，所以不能达到 PID 控制器实现的技术特性指标。利用 PID 设计程序，设 $K_I = 0$，即可完成 PD 控制器的设计。求得 PD 系数为 $K_P = 10$，$K_D = 0.68$。图 4-6 为系统补偿前后的阶跃响应。由图可见，超调量($\sigma\%$) = 24%，过渡过程时间 = 0.8s。误差常数 = 20。

由于微分器中的固有噪声问题，所以不可能构造一个纯微分环节($K_D s$)。然而，在许多情况下，微分项总有一个极点与它关联。在这种情况下，把 PD 控制器看作为超前补偿器(下面一节将详细讨论)。 PD 控制常可以实现直接对输出变化速度的测量(如转速)，见图 4-7。在这种情况下，微分项通常被放在被控对象的内环中。

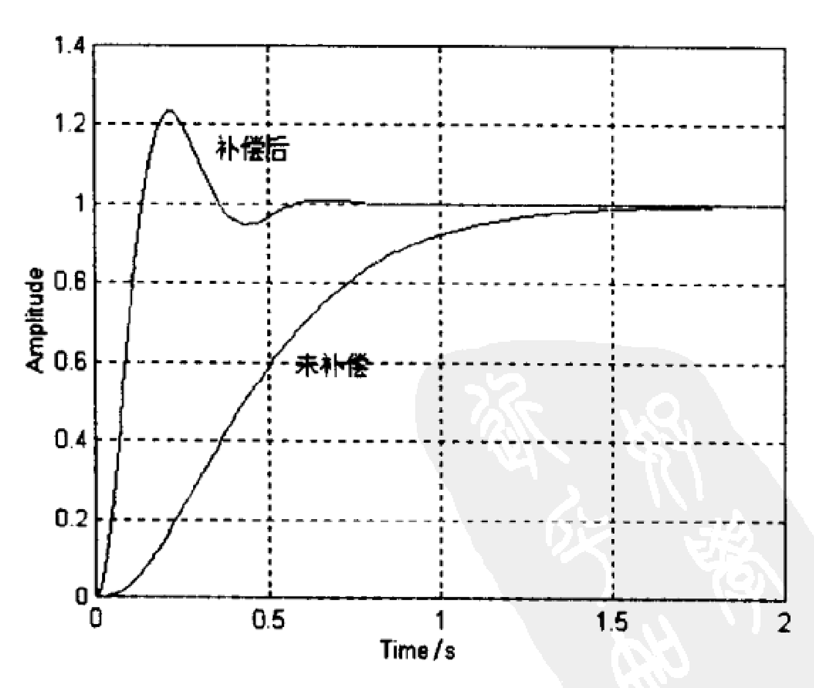


图 4-6　带有 PD 控制器的阶跃响应

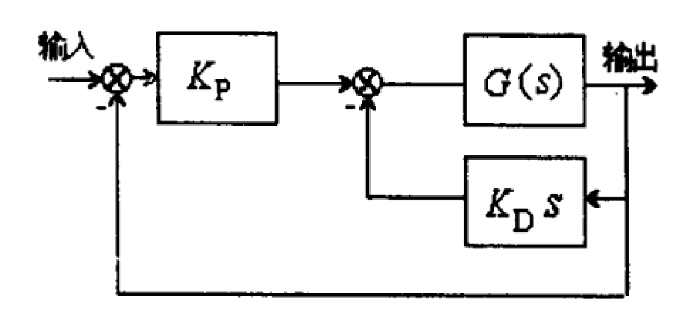


图 4-7　带有速度反馈的 PD 控制器结构图

系统的阶跃响应见图 4-8 。由图可见，系统超调量仅有 11%。这是由于微分项没有在反馈环中增加额外的零点。

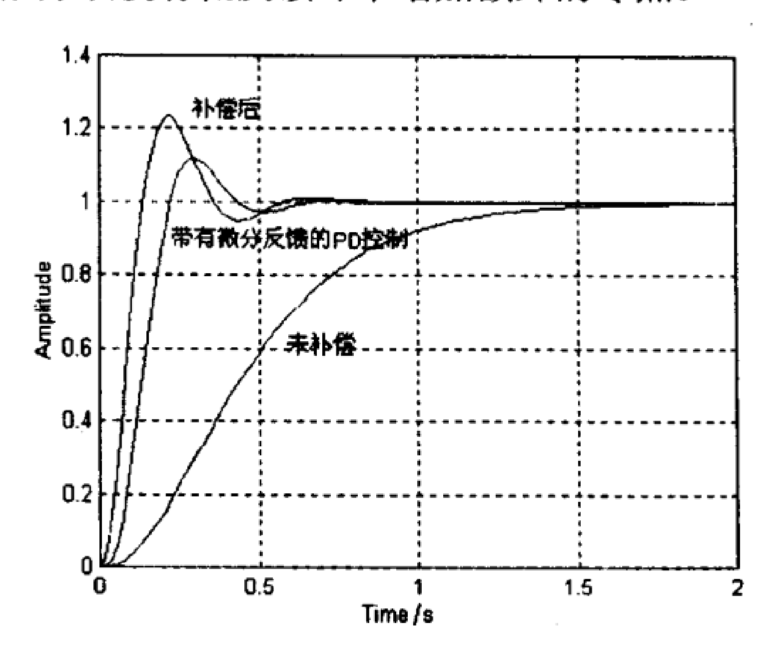


图 4-8　两种补偿方法阶跃响应的比较

4.4　超前补偿

最简单与最普通的补偿器形式是带有一个零点与一个极点的滤波器。对应的补偿器的传递函数为

$$K(s)=K_c\frac{s+a}{s+b}$$

如果零点出现在极点以前($0<a<b$)，$K(s)$即为超前补偿器的传递函数。如果极点出现在零点之前，$K(s)$即为滞后补偿器的传递函数。补偿器最大的相位补偿点(超前或滞后)为$\omega=\sqrt{ab}$。

设计者如何确定使用超前补偿或滞后补偿呢？当然，这要根

据具体的技术指标要求才能确定。例如，如果只要求闭环系统稳定，那么任一种补偿都可以被选用(假定被控系统是稳定的,对于不稳定系统,必须使用超前补偿)。然而，如再附加其它的性能指标要求，常常只能使用一种补偿器。例如，如果ω_{gc}是规定的，根据被控对象伯德图立刻可以确定补偿器的种类。如果希望的ω_{gc}大于被控系统的ω_{gc}，那么就需要超前补偿，如果小于被控系统的ω_{gc}，则需要滞后补偿。对于超前和滞后补偿器，可使用根轨迹、伯德图和解析方法进行分析设计。应该注意到，所有这些设计过程都建立在一对主导极点可以充分描述系统特性的前提下。实际的闭环系统响应，将与这些设计的结果总会有所不同，还需要在设计仿真后加入必要的调整。

所有的补偿器都影响闭环系统的稳定性、稳态误差与带宽。一般地讲，超前补偿使系统的相角裕量增加，从而提高系统的相对稳定性。对于给定的系统增益 K_c，超前补偿增加了系统的稳态误差。这是因为 $a<b$，则 $K_c a/b < K_c$。为减少稳态误差，必须使用大增益补偿器。超前补偿器也使增益穿越频率ω_{gc}增加。这样会使阶跃响应过渡过程时间减少(即增加系统阻尼)。ω_{gc}的增加，也会导致闭环系统的频带展宽,这也是我们希望的。系统频带的加宽也会引入不希望的噪声信号等。

如下设计过程可用于设计的初始工作。它仅仅可以作为一个指导性原则，不能作为一个定律或一般的方法，即不是按此原则设计的所有系统都可以得到充分的补偿。例如，应该限制补偿器的所有零极点具有负实部，以保证补偿器本身总是稳定的。但是在一些系统中，为了使闭环系统稳定，也需要不稳定的补偿器去补偿被控对象。利用稳定补偿器对一个本身可以稳定的系统进行补偿的系统称为强稳定系统。对于不能采用本章所讨论的经典方法使其稳定的系统，可以采用状态空间法或优化控制方法进行补偿。这些更先进的方法将在后面的章节中进行专门的讨论。

4.4.1 根轨迹设计方法

根轨迹设计方法，是通过给系统加入零点或极点而改变根轨

迹形状，追使根轨迹通过复平面上所希望的点的设计方法。

一般根轨迹设计步骤如下:

(1) 由给定的特征指标，确定希望的负极点 s_1 的位置。

(2) 确定补偿器零点位置。

(3) 利用根轨迹特征(s_1 必须位于根轨迹上； $K(s_1)G(s_1)$的角度必须是 180°)来确定补偿器的极点的位置。

(4) 此时系统在 s_1 点的增益值是否能使系统的稳态误差满足要求？如果不能满足的话，改变 s_1 并重复上面的设计。

(5) 系统闭环并仿真其输出特性，判别系统是否满足希望的特性要求。

有几种确定补偿器零点的方法。当然设计者可以凭他个人的经验与直觉来确定该零点的位置。有关如何确定补偿器零点的问题，请参阅有关自动控制系统设计方面的参考书。

在 S 平面上选择希望的极点 s_1，并求出

$$\varphi_c = 180° - \angle G(s_1)$$

所差的φ_c的角度由超前补偿器来补偿。如果φ_c为负值，就不能采用超前补偿器完成，而应改用滞后补偿器。一个超前补偿器所能补偿的角度约为 50°～ 60°。如果所需要的角度大，则应采用多级超前补偿器。其设计过程见图 4-9 。

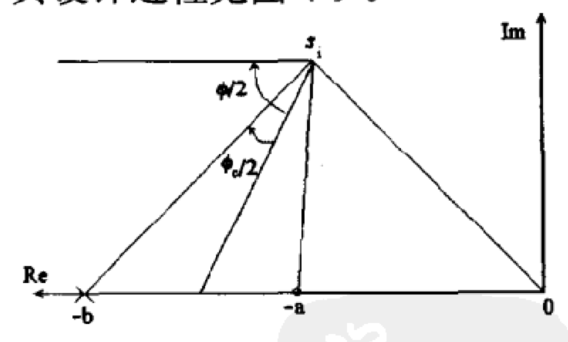


图 4-9　根轨迹几何设计方法

由坐标原点至 s_1 画一条线,并由 s_1 画一条水平直线。由这两条直线就可以求得角度φ。画角平分线。由希望零点与极点位置连线，使得每一条线与角平分线之间的夹角为$\varphi_c/2$ 。这样的设计过程使得零点与极点位置间的距离最小。通过保持比值系数接近

为 1，就会使得所求的保持器增益 K_c 值最小。

4.4.2 根轨迹的几何方法

从几何设计过程中,可以导出纯解析方法，并可容易地用 MATLAB 程序完成设计过程，见本章附录中程序 3 。

(1) 选择希望极点位置 s_1；

(2) 定义 $\varphi=\angle(s_1)$ 与 φ_c=180°−$\angle G(s_1)$；

(3) 定义 θ_p=（$\varphi-\varphi_c$）/2 和 θ_z=（$\varphi-\varphi_c$）/2；

(4) 由 $p_c=-b=\mathrm{Re}(s_1)-\mathrm{Im}(s_1)/\tan\theta_p$ 求得极点位置；

(5) 由 $z_c=-a=\mathrm{Re}(s_1)-\mathrm{Im}(s_1)/\tan\theta_z$ 求得零点位置；

(6) 求 K_c 使得 $|K(s_1)G(s_1)|=1$；

(7) 如果 K_c 太小，不能满足稳态误差要求,选择一个新的 s_1，并且重复上述过程。

【例 4-4】 设计超前补偿的几何方法。

作为设计过程的例子，重新检验例 4-1 中所讨论的被控系统传递函数。

$$G(s)=\frac{400}{s(s^2+30s+200)}$$

技术指标要求：

$$\zeta=0.5 \quad \omega_n=13.5\mathrm{rad/s}$$

由上述指标要求，可以导出在 s 平面上的希望极点的位置为 s_1=−6.75±j11.69 。ord2 命令可以被用来求对应二阶传递函数的 ζ 与 ω_n。描述上述设计求解过程如下：

$$\varphi=\angle(-6.75+\mathrm{j}11.69)=120°$$

$$\varphi_c=-180°-\angle G(-6.75+\mathrm{j}11.69)=55.8°$$

$$\theta_p=\frac{\varphi+\varphi_c}{2}=32°$$

$$\theta_z=\frac{\varphi+\varphi_c}{2}=87.9°$$

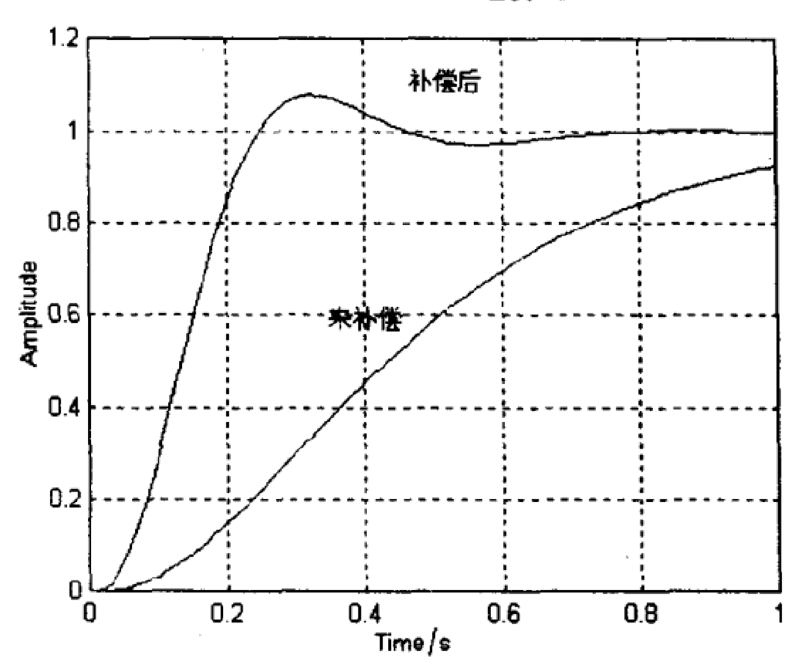
$$p_c=-b=\mathrm{Re}(s_1)-\mathrm{Im}(s_1)/\tan\theta_p=-25.41$$

$$p_z=-a=\mathrm{Re}(s_1)-\mathrm{Im}(s_1)/\tan\theta_z=-7.16$$

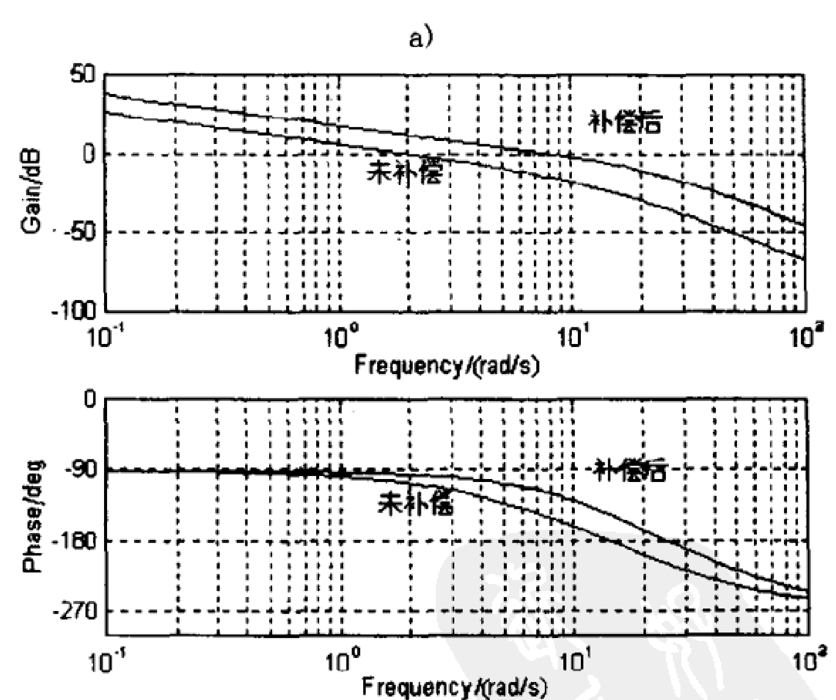
$$K_c = 1/|G(s_1)K(s_1)| = 13.62$$

最后补偿器传递函数为

$$K(s) = 13.62\frac{s+7.16}{s+25.41}$$



a)



b)

图 4-10 例 4-4 的阶跃响应与频率响应

a) 阶跃响应 b) 频率响应

系统的阶跃与频率响应见图 4-10 。由图可知,补偿后的系统超调量是 8%，过渡过程时间为 0.8s ，由伯德图可见，系统的增益裕量为 15dB，相角裕量为 60°，增益穿越频率为 8rad/s 。该结果与 PD 控制比较，是非常说明问题的。 PD 控制结果的超调量为 24%。而相角裕量为 45°。然而，两者主要不同点在于，使用超前补偿器设计得到的误差常数仅为 7.68，低于使用 PD 控制器设计得到的误差常数 20 。增益穿越频率(13.5rad/s, PD 控制)越高，系统的响应就越快。

另外，使用根轨迹法针对特定稳态误差的设计不直接。如果上述设计结果的稳态误差 7.68 不能满足要求，将需要再选择一个新的点，重复上述设计过程，直到达到可接受的稳态误差为止。然而，也有几种解析的方法使得设计者利用根轨迹法设计来满足稳态误差的特定要求。

4.4.3 根轨迹的解析方法

这里提出的解析方法，既可设计超前补偿器，也可以设计滞后补偿器。对于这种设计，可使用另外一种补偿器的表示法。

$$K(s)=K_{\mathrm{c}}\frac{s\tau_{\mathrm{z}}+1}{s\tau_{\mathrm{p}}+1}$$

首先，根据稳态误差与瞬态特征的要求，选定 K_{c} 与希望的 s_1 在 s 平面上的位置。补偿系统的根轨迹为

$$K(s_1)G(s_1)=K_{\mathrm{c}}\frac{s_1\tau_{\mathrm{z}}+1}{s_1\tau_{\mathrm{p}}+1}M_{\mathrm{G}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{G}}}=1\mathrm{e}^{\mathrm{j}\pi}$$

其中， $G(s_1)=M_{\mathrm{G}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{G}}}$ 。因为 K_{c} 已知，需要解上述方程来求 τ_{z} 与 τ_{p} 。如果 s_1 通过 $s_1=M_{\mathrm{s}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{s}}}$ 表示，那么

$$M_{\mathrm{s}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{s}}}\tau_{\mathrm{z}}+1=[\frac{1\mathrm{e}^{\mathrm{j}\pi}}{M_{\mathrm{G}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{G}}}K_{\mathrm{c}}}](M_{\mathrm{s}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{s}}}\tau_{\mathrm{p}}+1)$$

该方程可以分成实部与虚部两个部分，结果得到两个方程与两个未知数。这些方程的解为

$$\tau_{\mathrm{z}}=\frac{\sin\theta_{\mathrm{s}}-K_{\mathrm{c}}M_{\mathrm{G}}\sin(\theta_{\mathrm{G}}-\theta_{\mathrm{s}})}{K_{\mathrm{c}}M_{\mathrm{G}}M_{\mathrm{s}}\sin\theta_{\mathrm{G}}}$$

$$\tau_p = -\frac{K_c M_G \sin\theta_s + \sin(\theta_G + \theta_s)}{M_s \sin\theta_G}$$

如果τ_z与τ_p都是正的，该方法是可行的。当然，对于超前补偿器的设计，需要$\tau_p<\tau_z$。实际上，要经常选择几个K_c值进行设计，来选择其中最满足要求的补偿器参数，请见本章附录中程序清单 4。

【例 4-5】解析设计方法。

重复前面的系统设计，附带一项稳态误差要求，即

$$G(s) = \frac{400}{s(s^2+30s+200)}$$

技术指标要求如下：

$\zeta=0.5$，$\omega_n=13.5$ rad/s，速度误差常数=10

由技术指标要求可选$K_c=5$，$s_1=-6.75+\mathrm{j}11.69$

首先可以确定

$$s_1 = -6.75 + \mathrm{j}11.69 = 13.49\mathrm{e}^{\mathrm{j}\theta} = M_s\mathrm{e}^{\mathrm{j}\theta}$$

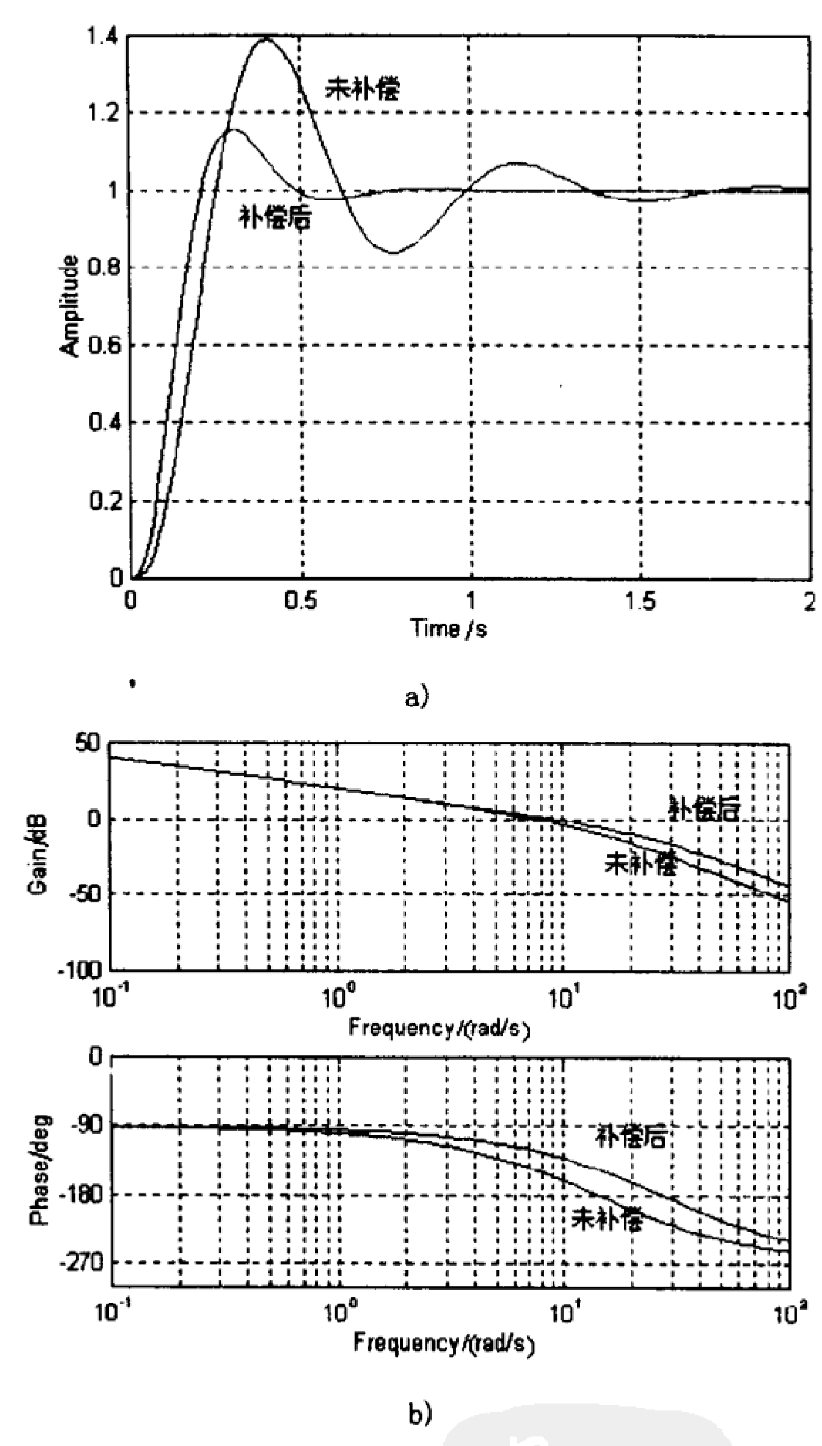
$$G(s_1) = 0.13\mathrm{e}^{\mathrm{j}124.1} = M_G\mathrm{e}^{\mathrm{j}\theta_G}$$

利用希望的K_c与上述参考值，求解τ_z与τ_p。对应该设计的补偿器为

$$K(s) = 5\frac{0.1s+1}{0.027s+1}$$

图 4-11 显示了该补偿器系统的阶跃与频率响应。补偿后的系统具有 14%的超调量与 0.9s 的过渡过程时间。该设计除已达到例 4-4 中设计的同样指标外，还具有较大的误差常数。为了比较，在图 4-11 中同时绘制了未补偿系统(增益=5)的曲线。由伯德图可见，使用补偿器可以增加稳定裕量、速度响应及减小超调。

如果K_c选为 10，则将会得到一个负的τ_p。一般地讲，为得到一个满意的设计，需要选择几个K_c值，反复执行设计程序来寻找较佳的设计结果。



a)

b)

图 4-11 例 4-5 中的阶跃响应与频率响应

a) 阶跃响应 b) 频率响应

4.4.4 超前补偿的伯德图设计方法

伯德图设计的基本思路是改变系统开环频率特性的形状，使其具有希望的低频增益(为满足稳态误差要求)、希望的增益穿越频率(为满足响应速度的要求)和充分的稳态裕量。伯德图设计中

常用的补偿器传递函数：

$$K(s)=K_{c}\frac{\alpha Ts+1}{Ts+1}$$

在伯德图设计方法中，首先选择增益的大小，以满足稳态误差要求。这也是伯德图设计方法的优点之一。然后，根据相角裕量的要求，求得希望的增益穿越频率。为得到更好的设计结果，应当搞懂所涉及到的公式推导过程。从补偿器所能补偿的相角与增益考虑，将很快地推导出一些适用的关系式。

$$K_{1}(s)=\frac{\alpha Ts+1}{Ts+1}$$

其中，$K(s)=K_{c}K_{1}(s)$

由于在前面根据稳态误差的要求，K_{c} 值已经选定，因此，在研究该问题时，仅考虑 $K_{1}(s)$的影响，即由 $K_{1}(s)$所提供的附加相角与幅值。超前补偿器的相位为

$$\angle K_{1}(\mathrm{j}\omega)=\arctan\alpha T\omega-\arctan T\omega$$

该相角是设计中使用的补偿器相角。因此我们更关心的是$\omega=\omega_{gc}$时相角大小。为了简单起见，通常在$\omega=\omega_{gc}$点处加入最大相位超前量，最大相位点的频率可以通过对上式求极值的方法求得。

$$\omega_{\max}=\frac{1}{\sqrt{\alpha}T}$$

最大相位超前量

$$\angle K_{1}(\mathrm{j}\omega_{\max})=\angle K_{1}(\mathrm{j}\omega_{gc})=\phi=\arctan\sqrt{\alpha}-\arctan\frac{1}{\sqrt{\alpha}}$$

利用三角学可以表明α与ϕ之间的关系：

当$\omega_{\max}=1/(\sqrt{\alpha}T)$时

$$\sin\phi=\frac{\alpha-1}{\alpha+1}$$

而

$$\alpha=\frac{1+\sin\phi}{1-\sin\phi}$$

我们注意到，在这种设计中，$\omega_{gc}=\omega_{max}$。由 $K_1(s)$引入的附加增益幅值(dB)为

$$M=\left|K_1(j\omega_{max})\right|_{dB}=\left|K_1(j\omega_{gc})\right|_{dB}=10\log\alpha$$

下面将详细讨论这个过程。

首先，使用满足稳态误差要求的 K_c 绘制伯德图，确定满足相角裕量要求的相角超前量，并计算α 值。由于现在已知超前补偿器将升高幅值 M(dB)，找出未补偿系统对应具有$-M$ dB 幅值的频率点。这样，迫使补偿后的系统在该频率点处为 0 dB 。该频率为一新的增益穿越频率，用它可以求得 T 值。该方法的设计步骤如下：

(1) 选择 K_c 达到满足需要的误差常数。

(2) 画出 $K_cG(j\omega)$的伯德图，并确定对应 $K_cG(j\omega)$的相角裕量 PM 。

(3) 确定需要的超前相位，这是初始的ϕ。

(4) 给初始的ϕ值加上几度作为下一步运算的ϕ值。

(5) 计算$\alpha=(1+\sin\phi)/(1-\sin\phi)$

(6) 找出 $K_cG(j\omega)$增益幅值为$-10\log\alpha$ 对应的频率点，该频率点即为ω_{gc}。

(7) 计算 $T=1/(\sqrt{\alpha}\omega_{gc})$

(8) 绘制 $K(j\omega)G(j\omega)$的伯德图，验证设计结果。

(9)使系统闭环，确定适当的闭环响应。

在设计过程中，为得到期望的相角裕量 PM ，需要在补偿相角上再加上几度（如 5°）的原因是，因为补偿器附加零点将会引起补偿系统穿越频率略有增加。

我们也注意到，在这种设计方法中，不能事先确定补偿的角频率ω_{gc}。如果ω_{gc}不能满足要求，就不得不调整相角裕量与稳态误差的要求。

【例 4-6 】伯德图方法。

被控对象传递函数为

$$G(s)=\frac{400}{s(s^2+30s+200)}$$

设计技术指标要求如下：

速度误差常数=10

PM=45°

由速度误差常数的要求可得 K_c=5 ， $K_cG(j\omega)$的伯德图见图 4-12 。由图可见，此时系统相角稳态裕量为 32°。因此，需要再补偿ϕ=13°。在计算α时,再将ϕ加上 5°裕量，得α=1.89 。由此又可以得到$-10\log\alpha=-2.77$dB 。由图 4-12 可以看出，ω_{gc}=9 rad/s ，因此

$$T=\frac{1}{\sqrt{\alpha}\omega_{gc}}=0.08$$

补偿器传递函数为

$$K(s)=5(0.15s+1)/(0.08s+1)$$

补偿后系统 PM=41°，接近设计要求。系统阶跃响应见图 4-13 。由图可见，超调量为 28%，而过渡过程时间为 1.4s 。

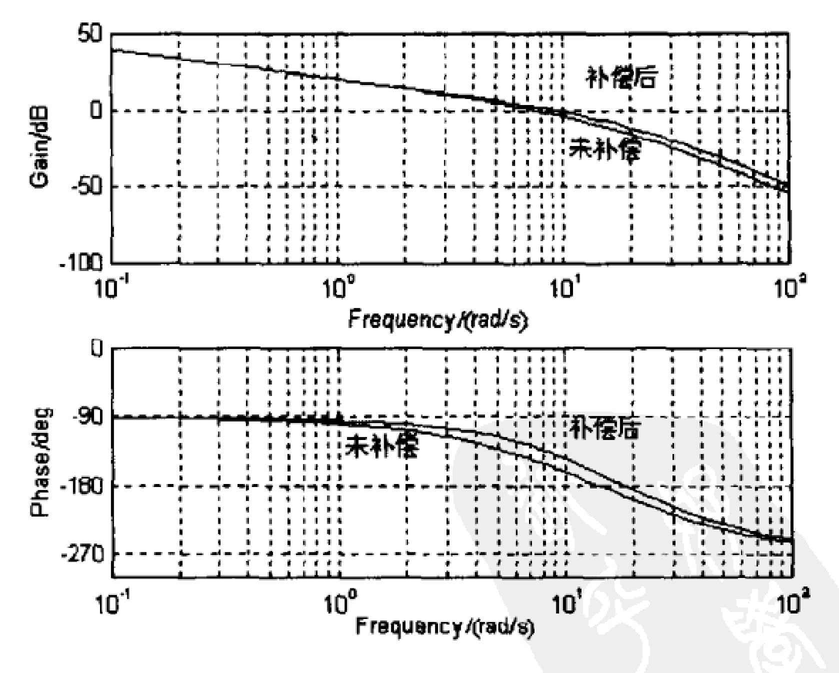


图 4-12　例 4-6 系统的伯德图

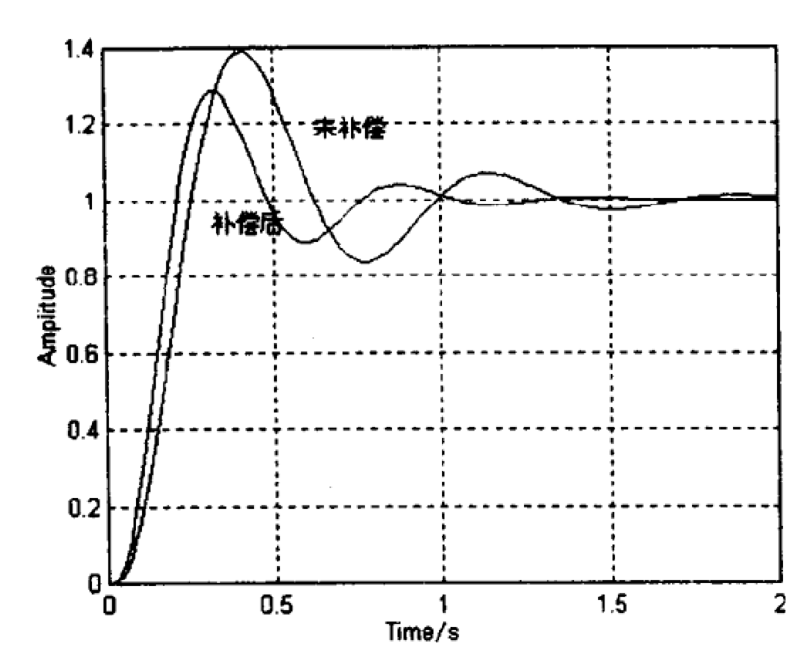


图 4-13　例 4-6 系统的阶跃响应

4.4.5　伯德图设计的解析方法

前面介绍的根轨迹解析方法修改后可用于伯德图设计。在这种情况下，希望补偿后的系统 $K(s)G(s)$增益为 1.0 ，而在 $s=\mathrm{j}\omega_{\mathrm{gc}}$ 点处相角为-180°+PM 。假定补偿器的时间常数为已知，可以得到

$$K(\mathrm{j}\omega_{\mathrm{gc}})G(\mathrm{j}\omega_{\mathrm{gc}})=K_{\mathrm{c}}\frac{\mathrm{j}\omega_{\mathrm{gc}}\tau_{\mathrm{z}}+1}{\mathrm{j}\omega_{\mathrm{gc}}\tau_{\mathrm{p}}+1}M_{\mathrm{G}}\mathrm{e}^{\mathrm{j}\theta_{\mathrm{G}}}=1\mathrm{e}^{\mathrm{j}(-180+\mathrm{PM})}$$

式中，M_{G} 与 θ_{G} 是 $G(\mathrm{j}\omega)$在 $\omega=\omega_{\mathrm{gc}}$ 点的增益与相角。该方程可以分为实部与虚部两部分。因此可以写出具有两个未知数的两个方程。求解该方程组可得

$$\tau_{\mathrm{z}}=\frac{1+K_{\mathrm{c}}M_{\mathrm{G}}\cos(\mathrm{PM}-\theta_{\mathrm{G}})}{-\omega_{\mathrm{gc}}K_{\mathrm{c}}M_{\mathrm{G}}\sin(\mathrm{PM}-\theta_{\mathrm{G}})}$$

$$\tau_{\mathrm{p}}=\frac{\cos(PM-\theta_{\mathrm{G}})+K_{\mathrm{c}}M_{\mathrm{G}}}{\omega_{\mathrm{gc}}\sin(\mathrm{PM}-\theta_{\mathrm{G}})}$$

为利用这些方程，首先确定 K_{c}，并绘制 $K_{\mathrm{c}}G(\mathrm{j}\omega)$伯德图。由图可以得到$\omega=\omega_{\mathrm{gc}}$ 点处的 K_{c}、 M_{G} 及θ_{G}。注意，此时 $K_{\mathrm{c}}M_{\mathrm{G}}$ 是实际幅值，而不是以 dB 为单位的。由附录的程序 6 可以完成这个设计运算过程。

【例 4-7】伯德图解析方法。

已知

$$G(s)=\frac{400}{s(s^2+30s+200)}$$

设计技术指标要求如下：

单位斜坡输入的稳态误差小于 10%；

ω_{gc}=14 rad/s

PM=45°

首先考虑满足稳态误差的要求，由此得到 K_c=5 。如图 4-12 所示，在 $\omega=\omega_{gc}$=14rad/s 点处

$$K_cM_G=0.34 \qquad \theta_G=-180°$$

利用这些值与希望的 ω_{gc} 和 PM ，可得到 τ_z=0.227, τ_p=0.038 。

因此，补偿器传递函数为

$$K(s)=5\frac{0.227s+1}{0.038s+1}$$

补偿后的系统伯德图见图 4-14 。可以看到，ω_{gc} 与 PM 指标的要求已得到满足。系统闭环阶跃响应曲线见图 4-15 。其超调量与过渡过程时间分别为 19%和 0.9s 。

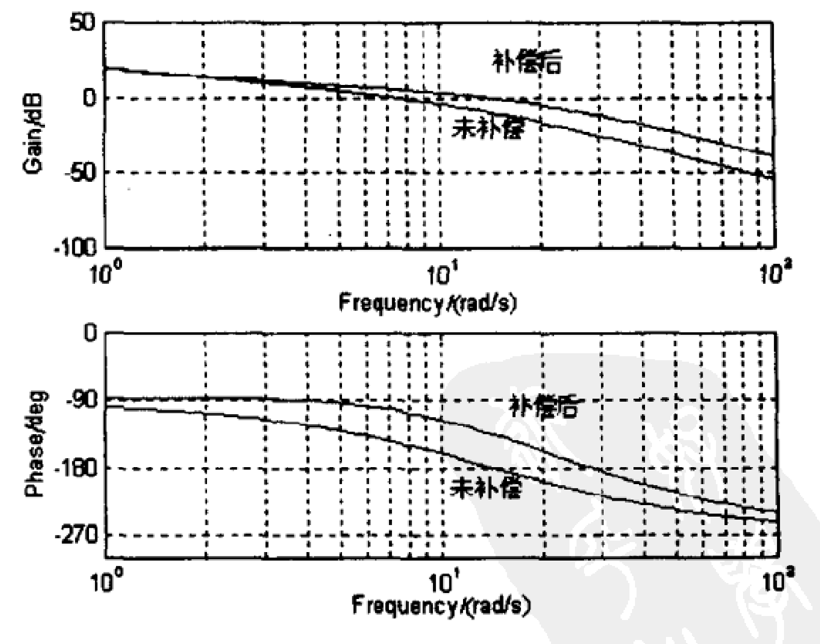


图 4-14 例 4-7 的伯德图

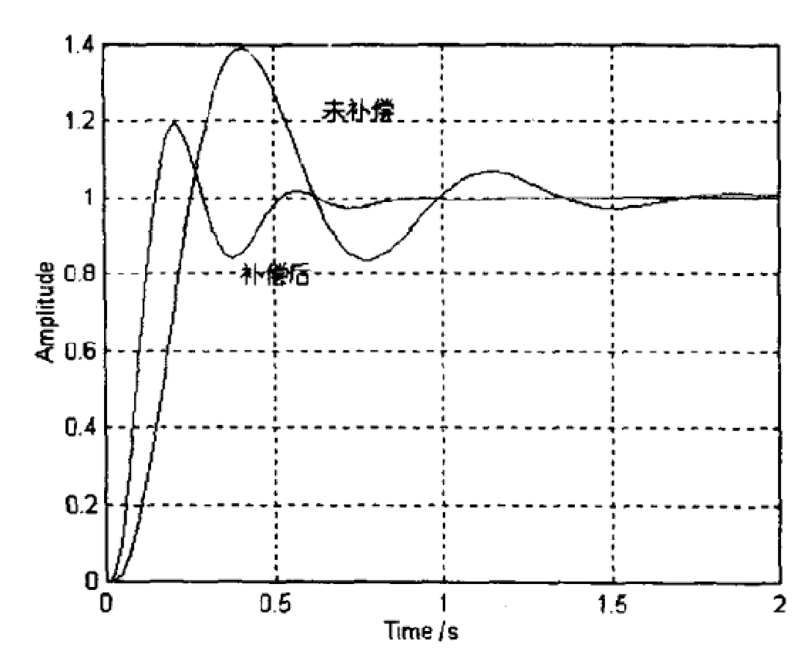


图 4-15 例 4-7 的系统阶跃响应

4.4.6 PD 控制器与超前补偿器的比较

在 4.4.3 节中讨论的 PD 控制器在实际中是不可能实现的。微分环节会严重降低信号与噪声的比值，因此，应该避免使用微分环节。实际 PD 控制器的传递函数为

$$\mathrm{PD}(s)=K_{\mathrm{P}}+\frac{K_{\mathrm{D}}s}{sT_{\mathrm{d}}+1}$$

如写成一个单分数，则为

$$\mathrm{PD}(s)=\frac{K_{\mathrm{P}}T_{\mathrm{d}}s+K_{\mathrm{P}}+K_{\mathrm{D}}s}{T_{\mathrm{d}}s+1}=K_{\mathrm{P}}\frac{(T_{\mathrm{d}}+K_{\mathrm{D}}/K_{\mathrm{P}})s+1}{T_{\mathrm{d}}s+1}$$

上式即为超前补偿器的公式。因此 PD 控制与超前补偿器的作用是相似的。在理想情况下，微分控制器相当于超前环节的极点为无穷大的情况。

4.5 滞后补偿

在滞后补偿器中，极点要小于零点。因为滞后补偿器给系统加入了滞后相角，因此使系统趋于不稳定。对应这个原因，如果原系统已经不稳定或稳定裕量很小，就不要使用滞后补偿。然而，有时滞后补偿可以通过降低系统增益而提高系统的相对稳定性。后面将详细讨论这个问题。

滞后补偿器基本作用之一是，降低系统稳态误差。考虑滞后补偿器极点趋向于零的极限情况，即可容易说明这一作用。在这种情况下，滞后补偿器相当于一个积分器，添加到系统中使得系统增型，因此可降低系统稳态误差。对于一般的情况，滞后补偿器中 $a>b$ ，因此 $K_c a/b>K_c$ ，使得系统稳态误差减小。

滞后补偿使得系统 ω_{gc} 降低，从而使得系统响应变慢。相反地，需要经常限制闭环系统的带宽，以限制外部噪声信号进入控制环。在这种情况下，希望使用滞后补偿来降低闭环系统的带宽。

4.5.1 根轨迹设计方法

为讨论方便起见，仍然使用上述补偿器传递函数

$$K(s)=K_c\frac{s+a}{s+b}$$

根轨迹设计方法常被用到要满足系统稳态误差要求，而此时系统已经具有满意的稳定裕量与动态性能的系统设计中。在这种设计方法中，其设计过程如下：

(1) 根据给定的特性，确定 s_1 在 s 平面的位置；

(2) 绘制被控对象的根轨迹 $G(s)$；

(3) 在 s 平面上确定希望点上的 K_c 值；

(4) 如果 K_c 太小，不能满足稳态误差要求，进入下一步；

(5) 选择 a/b 的比值，以满足希望的稳态误差；

(6) 保持该比值，安置极零点，以达到希望的根轨迹；

(7) 核查系统闭环响应。

上述(6)项的设计是一种技巧。没有一定的规则保证设计成功。然而，下述的方法却被广泛地使用。选择 a 与 b 相互靠近，并且要保持所需要的比值。如何使两个数非常靠近，而又使其保持需要的比值(如等于 10)？回答是在 0 至 1 之间选择它们。例如，如果 b=0.01 而 a=0.1 ，则它们的比值为 10 。因为它们几乎可以对消，因此它们对系统的时域响应的影响有时可以被忽略。其对时域响应的主要负作用是小的补偿零点会吸引其中一个极点。因此系统将有一个靠近原点的闭环极点。这将对系统的过渡过程时

间影响很大。这个极点越小，过渡过程时间就越长。

【例 4-8】滞后补偿器的根轨迹设计法。

被控对象传递函数为一个 I 型系统。

$$G(s)=\frac{10}{s(s+5)}$$

设计技术指标要求如下：

阻尼系数ζ=0.707

单位斜坡稳态误差小于 5%

在 s 平面根轨迹上，满足阻尼系数 0.707 的点为

$$s_1=-2.5+2.5\mathrm{j}$$

在这一点的 K_c 值为 1.25 。对应此值的误差常数为 2.5 ，稳态误差为 40%。

为降低稳态误差，使用 a/b 比值为 8 的滞后补偿器。设定补偿器零点，a=0.1 ，而由计算可得 b=0.0125 。补偿器变为

$$K(s)=1.25\frac{s+0.1}{s+0.0125}$$

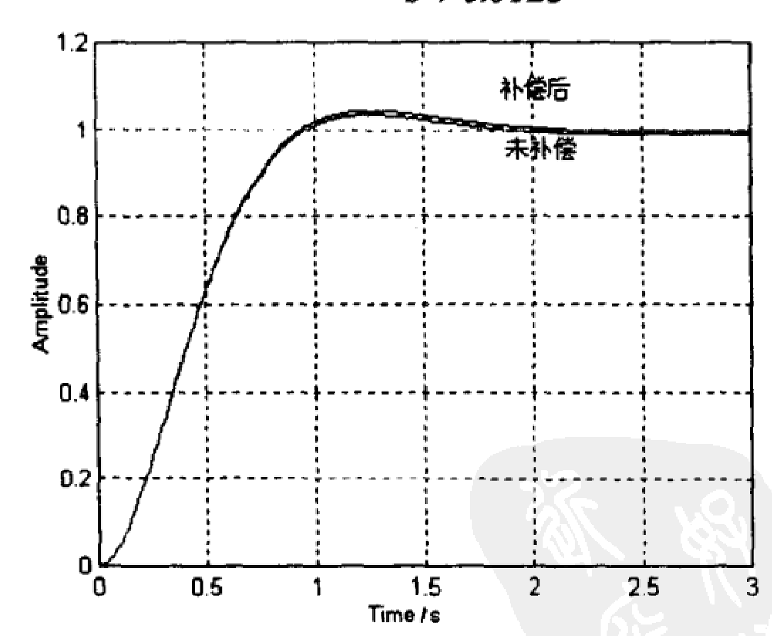


图 4-16　例 4-8 的阶跃响应

闭环极点与零点分别为{−0.1037,−2.45±j2.45}与{−0.1}。由零极点分布可看出，闭环系统的实零点与实极点可以对消。因此，复数极点可为主导极点，并满足阻尼比要求。补偿前后的系统闭

环阶跃响应见图 4-16 。未补偿系统的阶跃响应曲线 K_c=1.25 。该响应虽满足阻尼比要求，但是有较大的稳态误差。补偿后的阶跃响应比补偿前的系统阶跃响应超调量稍大一些，过渡过程时间稍长一些，但是它满足了系统稳态误差指标。

4.5.2 根轨迹的解析方法

前面已经讨论过在超前补偿器设计中，使用根轨迹数据的解析方法。从理论上讲，前面的方法也适用于滞后补偿器的设计，可在方程中直接地插入相关的数据。在这种情况下，可发现$\tau_p>\tau_z$。上述问题就是典型滞后补偿器设计的基本原理。在这种情况下，原来系统的根轨迹是可以接受的，仅仅使用了滞后补偿器来增加系统稳态误差常数。基于在 s 平面上$\theta_G\approx180°$的原理。可以求出τ_p与τ_z。

然而，如果打算利用滞后补偿器改变ζ或ω_n，那么可以按如下步骤进行：

(1) 选择 K_c 满足稳态误差的要求；

(2) 绘制 $K_cG(s)$根轨迹；

(3) 根据操作性能要求，确定 s 平面上 s_1 的位置；

(4) 找出 M_s 与θ_s，其中 $s_1=M_s e^{j\theta_s}$；

(5) 由 $K_cG(s_1)=K_cM_G e^{j\theta_G}$ 找出 K_cM_G 与θ_G，然后计算

$$\tau_z=\frac{\sin\theta_s-K_cM_G\sin(\theta_G-\theta_s)}{K_cM_GM_s\sin\theta_G}$$

$$\tau_p=-\frac{K_cM_G\sin\theta_s+\sin(\theta_G+\theta_s)}{M_s\sin\theta_G}$$

(6) 绘制 $K(s)G(s)$根轨迹图，验证设计结果；

(7) 使系统闭环，求解系统时域响应。

【例 4-9】 滞后补偿器的根轨迹解析设计方法。

被控对象传递函数为

$$G(s)=\frac{10}{s(s+5)}$$

设计的技术指标要求如下：

单位斜坡稳态误差小于 5%；

闭环阻尼比ζ=0.707，ω_n=1.5 rad/s。

由稳态误差要求可得 K_c=10，对应要求的阻尼比与ω_n、s平面上的对应点为

$$s_1=-1.06+j1.06$$

与其相关的幅值与相角为

$$M_s=1.5,\ \theta_s=135° \quad M_G=1.63, \quad \theta_G=-150°$$

最后，可得到τ_z=1.232，τ_p=15.091，因此补偿器传递函数为

$$K(s)=10\frac{1.232s+1}{15.091s+1}$$

对应闭环系统极点为{$-1.06\pm j1.06$，-2.94}与零点{-0.81}

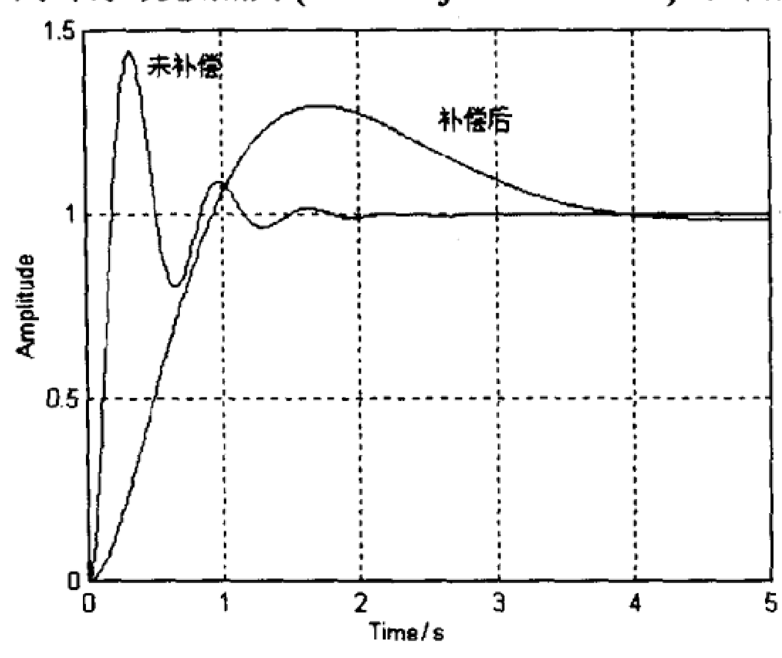


图 4-17　例 4-9 系统的阶跃响应

补偿前后的系统阶跃响应与系统伯德图见图 4-17 与图 4-18。在前面的例题中，显示的未补偿系统的时域响应满足要求。在本例题中，未补偿系统响应带有附加的增益 10。因此它满足稳态误差指标。然而应注意到，此时的未补偿器超调量达 43%，而补偿后的系统超调量为 30%。由系统伯德图可见，系统相角裕量 PM 由 28°增加到 48°。因此增益穿越频率由 9.4rad/s 减少到

1.7rad/s 。

注意到，即使阻尼系数为 0.707，但是超调量已高于希望值。这是由于闭环系统零点(来自补偿器零点)位于复数极点右方，从而使其主导极点作用消失的原因。如果超调量过大，补偿器零点应该向左移，直至抵消掉闭环系统的实极点。问题是该极点的位置不能提前确定，因此应该采用试探法或其它方法。

4.5.3 滞后补偿器的伯德图设计方法

由图 4-18 所示的伯德图可见，滞后补偿器减小了系统增益，并且添加滞后相角。当希望减少被控对象增益时，通常采用滞后补偿。这样做的结果或是增加相角裕量，或是降低闭环系统带宽。正如超前补偿器设计方法一样，最普通的伯德图方法使得设计工程师可根据稳态误差与希望相角裕量要求来设计滞后补偿器。在这里，将讨论一种伯德图设计方法。有关纯解析方法将在下面一节中讨论。

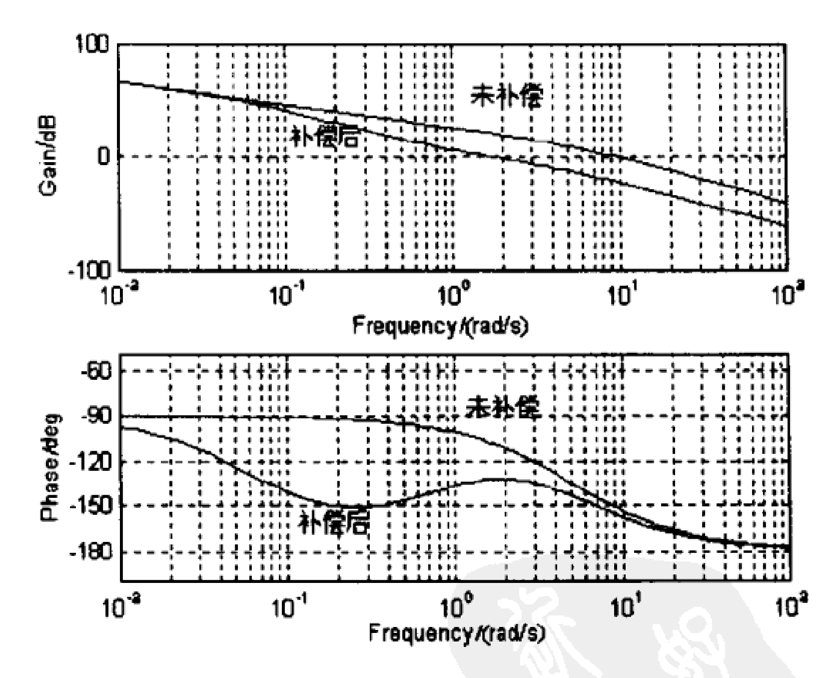


图 4-18　例 4-9 系统的伯德图

对于该方法，滞后补偿器最普通形式为

$$K(s)=K_c\frac{1+\alpha T_s}{1+T_s}\quad(\alpha<1)$$

如在超前补偿器设计中一样,首先选择 K_c 来满足系统稳态误差的要求。然后寻找 α 与 T 参数来满足需要的相角裕量。在设计之前，首先分析这些参数是如何影响补偿器增益与相角的。计算补偿器在 $\omega \to \infty$ 时最大增益衰减值为

增益衰减　　　　GR(dB)=20log α

也可以得到在 $\omega=10/(\alpha T)$处，滞后补偿器有一最小相位点。根据这些情况，可引出下述设计方法：

(1) 确定 K_c 值满足稳态误差要求；

(2) 绘制 $K_cG(j\omega)$的伯德图；

(3) 如果相角裕量不足,找出满足相角裕量 PM(再加 5°裕量)的频率点。该频率将作为补偿系统的ω_{gc}；

(4) 求得 $K_cG(j\omega)$在 $\omega=\omega_{gc}$ 点处的增益，该增益量将是由滞后补偿器消减的增益量值，即

$$\mathrm{GR} = -\left|K_cG(j\omega_{gc})\right|_{dB} \to \alpha = 10^{\mathrm{GR}/20}$$

(5) 补偿器最小相位作用点

$$T = \frac{10}{\alpha\omega_{gc}}$$

(6) 绘制 $K(j\omega)G(j\omega)$伯德图，检验设计结果；

(7) 仿真闭环系统。

附录中程序清单 7 可以用来求解下述例题。

【例 4-10】滞后补偿器的伯德图方法。

同例 4-9 中系统与性能指标要求。

首先选择 K_c=10 来满足系统稳态误差的要求。然后绘制 K_c=$G(j\omega)$伯德图，见图 4-19 。

若闭环系统阻尼比为 0.707，则需要相角裕量 PM 为 70°(这是由经验近似公式 PM=100 ζ而得)。再加入 5°裕量，并检查由伯德图确定的满足相角裕量 PM 的频率点，可以得到补偿后的系统$\omega_{gc} \approx$ 1.34rad/s 。由于 $K_cG(j\omega_{gc})$的增益为 23dB，可以求得

$$\alpha=0.069 \qquad T=106.7$$

因此，滞后补偿器的传递函数为

$$K(s)=10\frac{7.46s+1}{106.7s+1}$$

其闭环极点与零点为$\{-0.148,-2.43 \pm j0.64\}$和$\{-0.134\}$。系统伯德图与阶跃响应见图 4-19 与图 4-20 。由图可见，超调量由43%降至 7%，增益穿越频率为 1.35rad/s ，相角稳定裕量 PM 为70°。

注意，该补偿器滞后作用点经常出现在低频处。大多数这种滞后补偿的作用在接近系统穿越频率时会消失。顺便在这里可以解释这就是为什么在前面根轨迹设计法中选择小数极点与零点的原因，以保证滞后相角出现在足够低的频率段，使其不影响系统的相角裕量。

在比较以上根轨迹与伯德图设计方法的例题中，可以看到，没有明显的选择界限。对于该系统被控对象与设计特性要求，根轨迹法的设计给出了较快速的时域响应结果，而伯德图法的设计给出了较小超调量的设计结果。对于任何给定的问题，可使用两种方法进行设计，然后选择最接近要求的闭环时域响应的设计结果。

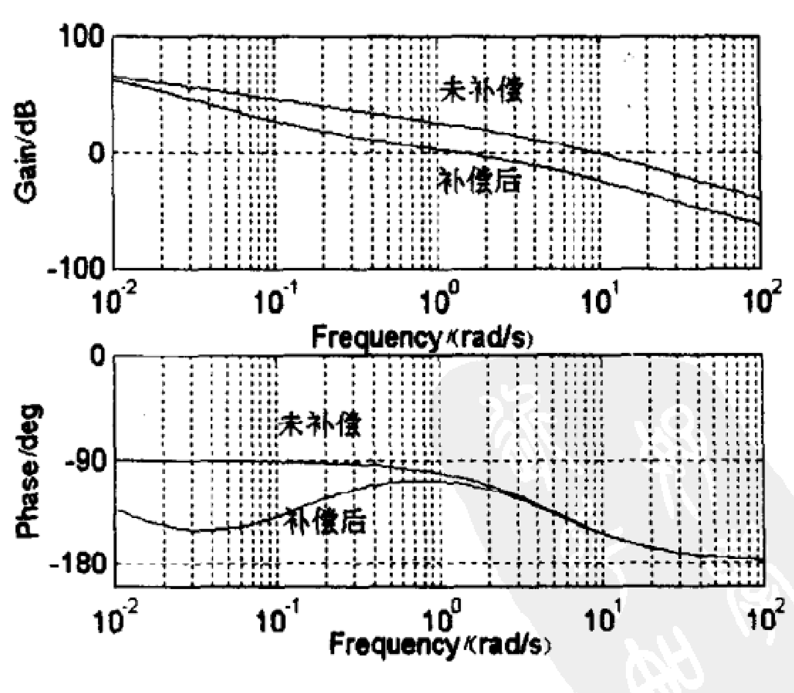


图 4-19　例 4-10 系统的伯德图

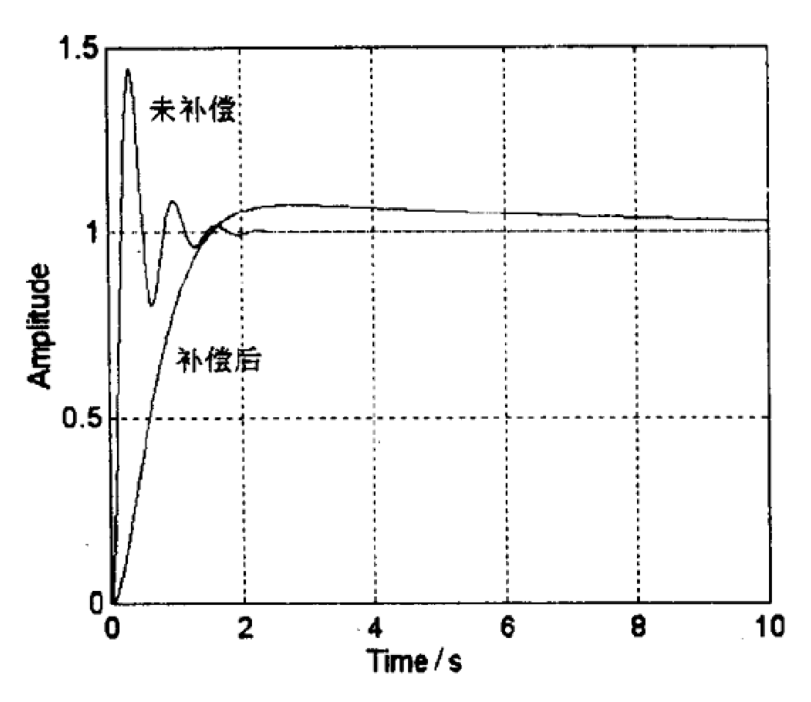


图 4-20 例 4-10 的系统阶跃响应

前面介绍的设计方法的一个缺点是，不能够任意选择一个希望的增益穿越频率。于是滞后补偿器通常用于调整稳态误差；超前补偿可以提高系统其它性能指标。然而，在某些情况下，使用滞后补偿器设计即可达到稳态指标要求，也可满足瞬态指标要求。

4.5.4 伯德图设计的解析方法

前面讨论的伯德图超前补偿器设计的解析方法，再加一些限制即可以用于滞后补偿器设计。首先重复上述设计规则，给出一个例子，然后再讨论该种方法的限制。

(1) 选择 K_c 满足稳态误差的要求；

(2) 绘制 $K_cG(\mathrm{j}\ \omega)$ 的伯德图，在希望的 $\omega=\omega_{gc}$ 点处确定 K_c、M_G 和 θ_G；

(3) 对于希望的 PM，由下式求出极点与零点的时间常数：

$$\tau_z = \frac{1+K_c M_g \cos(\mathrm{PM}-\theta_G)}{-\omega_{gc} K_c M_G \sin(\mathrm{PM}-\theta_G)}$$

$$\tau_p = \frac{\cos(\mathrm{PM}-\theta_G)+K_c M_G}{\omega_{gc}\sin(\mathrm{PM}-\theta_G)}$$

(4) 绘制补偿后的伯德图，验证设计结果；

(5) 仿真闭环响应。

【例 4-11】 滞后补偿器的伯德图解析方法。

下列为被控系统的传递函数与特性指标要求：

$$G(s)=\frac{10}{s(s+5)}$$

单位斜坡输入稳态误差小于 5%。

ω_{gc}=2rad/s， PM=40°

首先设定 K_c=10，并绘制 $K_cG(j\omega)$的伯德图，利用本章附录中程序 6，可得到

$$M_G=0.92 \qquad \theta_G=-111.8°$$

补偿器的参数为 τ_z=0.81， τ_p=8.89 。

补偿器传递函数为

$$K(s)=10\frac{0.81s+1}{8.89s+1}$$

该闭环系统的极点与零点分别为{–0.98±j1.61 ，–3.1430}与{–1.22}。

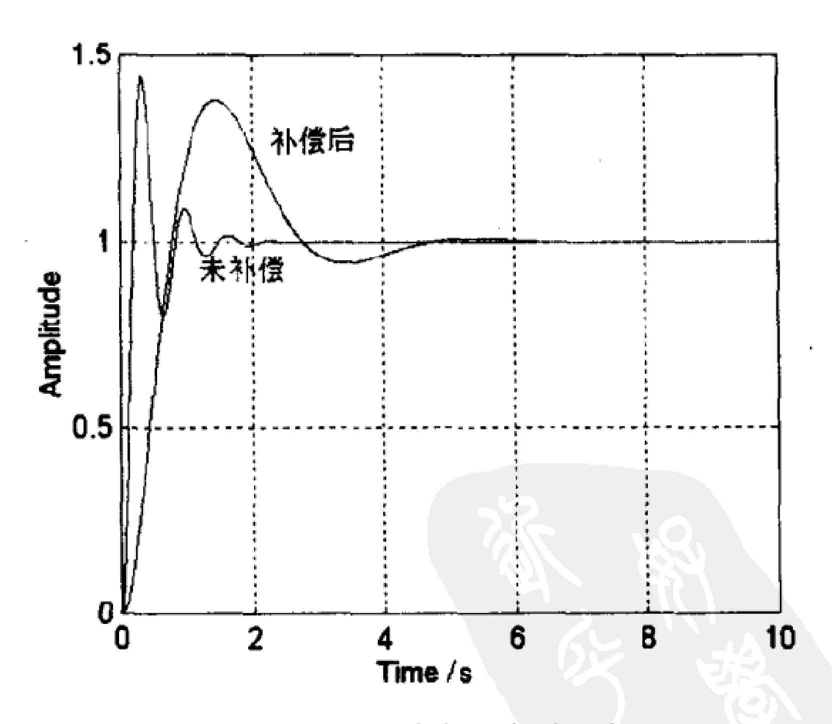


图 4-21　例 4-11 系统的阶跃响应

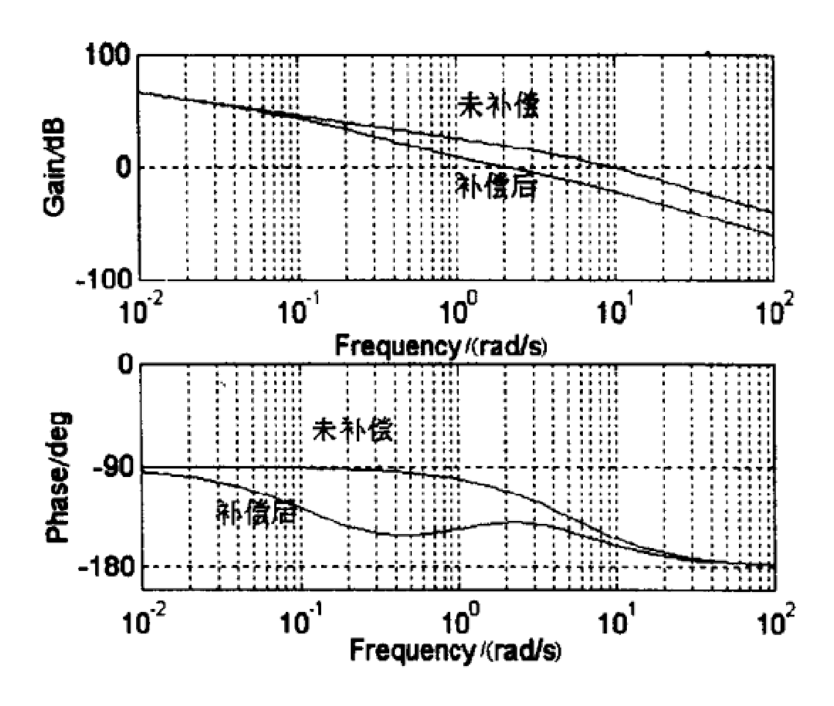


图 4-22 例 4-11 系统的伯德图

由阶跃响应曲线与伯德图(见图 4-21 与图 4-22)可见，系统满足要求的特性指标。大超调的阶跃响应是由于复数极点的低阻尼率所致。

我们可以修改设计程序,使用循环与条件语句计算一套补偿器参数，并绘制它的阶跃响应，然后选择其中最能满足性能指标要求的设计结果作为补偿器的参数。另外还可以使用 MATLAB 工具箱中更高级的优化算法进行优化设计。

4.5.5 PI 控制器与滞后补偿器的比较

比例积分控制器传递函数为

$$K_P + \frac{K_I}{s} = \frac{K_P s + K_I}{s} = K_P \frac{s + K_I / K_P}{s}$$

由此可见， PI 控制器是滞后补偿器的特殊情况。此时补偿器极点全移到坐标原点。

4.6 一般补偿控制

对于许多系统，单使用讨论过的设计方法，如 PID 控制、超前补偿或滞后补偿，不能达到要求的性能指标。在有些情况下，可以使用两种或更多控制或补偿方法。在另外情况下，设计者常

采用一种特定的设计方法，即设计者利用个人知识与经验选择补偿器的极点与零点位置，利用计算机仿真其设计结果，重复其设计过程直至达到可以接受的结果为止。在经典控制系统设计中，设计者的设计技巧、设计经验与理论知识，对设计工作是非常重要的。

作为本章最后一个例题，讨论一个直升飞机的控制问题。

【例 4-12】 直升飞机在盘旋时纵向运动的状态空间模型为

$$\begin{bmatrix}\dot{q}\\ \dot{\theta}\\ \dot{v}\end{bmatrix}=\begin{bmatrix}-0.4 & 0 & -0.01\\ 1 & 0 & 0\\ -1.4 & 9.8 & -0.02\end{bmatrix}\begin{bmatrix}q\\ \theta\\ v\end{bmatrix}+\begin{bmatrix}6.3\\ 0\\ 9.8\end{bmatrix}\delta$$

$$y=\begin{bmatrix}0 & 0 & 1\end{bmatrix}\begin{bmatrix}q\\ \theta\\ v\end{bmatrix}$$

式中，q 为倾斜率；θ 为机壳倾斜角；v 为水平速度；δ 为旋翼倾斜角。

利用 ss2tf 与 printsys 命令将其转换为传递函数形式并显示。

```
》[ng， dg]=ss2tf(a， b， c， d);
》 printsys(ng， dg)
        num / den =
```

$$\frac{9.8s\text{^}2-4.9s+61.74}{s\text{^}3+0.42s\text{^}2-0.006s+0.098}$$

以上给出了右旋翼倾斜角度与水平速度之间的传递函数。

$$G(s)=\frac{9.8s^2-4.9s+61.74}{s^3+0.42s^2-0.006s+0.098}$$

该系统在右平面上既有零点又有极点，由 ss2zp 命令可得

```
》[open_zer， open_pol， gain]=ss2zp(a,b,c,d)
open_zer=
          0.2500+2.4975i
          0.2500-2.4975i
open_pol=
          -0.6565
```

```
        0.1183+0.3678i
        0.1183-0.3678i
gain=
       9.800
```

很明显，该系统对应任何增益都是不稳定的。利用前面讨论的任何方法都不能使系统稳定。因此要采用特定的设计方法。

首先，使用 $K_1(s)=(s+0.65)/(s+10)$抵消掉在(−0.65, j0)处极点。该极点对系统稳定性的影响是很大的，因此在离它约 10 倍远的地方添加另外一个极点。下一步在原点处放置一个零点，以吸引右半平面的极点进入左半平面。也需要添加另外一个极点，下面试探性地给出补偿器传递函数

$$K(s)=K_c\frac{s+0.65}{s+10}\times\frac{s}{s+10}$$

使用 series 命令可实现多级传递函数的串联。 minreal 命令可以用来抵消传递函数中相同的极点与零点。最后可以绘制系统根轨迹。补偿后系统在原点处的根轨迹见图 4-23 。该系统是一个条件稳定系统，即如果增益过高或过低都会使得系统变得不稳定。当 K_c=2 时，系统稳定裕量最大(可由 rlocfind 命令求得这个结果)。该特殊设计是采用了两个超前补偿器串联的方法。

最后，使用 cloop 命令使系统闭环，并求得系统的阶跃响应，见图 4-24 。由图可见，由于原点处补偿器零点的存在，系统输出随时间的增长趋向于零。如果这个阶跃输入信号是一干扰信号，那么这种输出响应结果正是我们所希望的。

然而，在这种情况下，希望用旋翼倾斜角度控制其纵向速度，可以将补偿器分为两个部分实现这种控制。补偿器可以分为

在前向通道中

$$K_1(s)=\hat{k}_1\frac{s+0.65}{s+10}$$

在反馈通道中

$$K_2(s) = \hat{k}_2 \frac{s}{s+10}$$

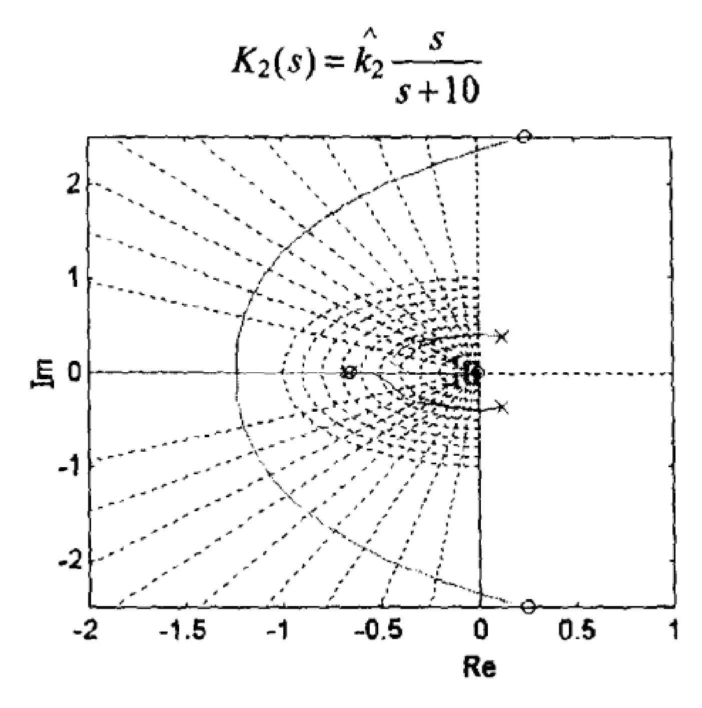


图 4-23　例 4-12 补偿系统的根轨迹图

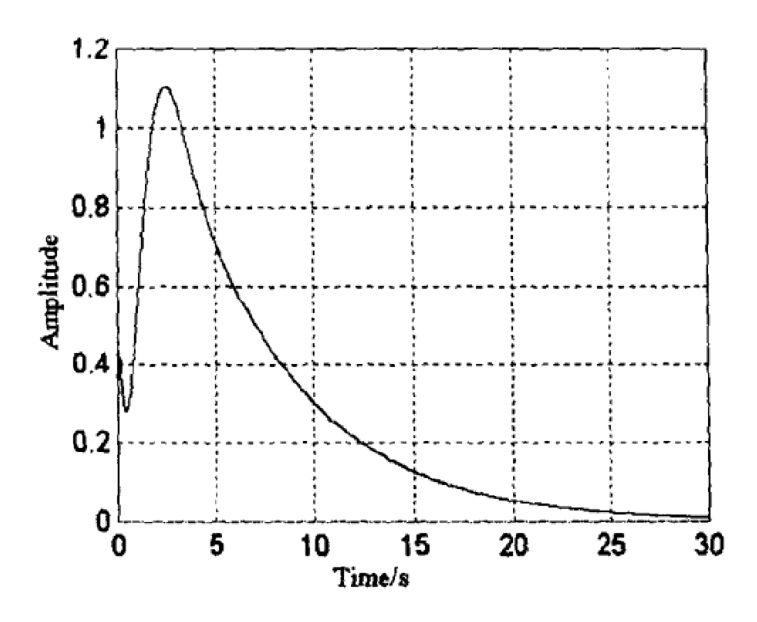


图 4-24　例 4-12 的系统阶跃响应

注意到 $K_c = 2 = \hat{k}_1 \hat{k}_2$，现在选择增益值，以实现对单位斜坡响应的零稳态误差。这意味着，在零频率点处闭环传递函数必须为 1，即 $T(0)=1$，其中

$$T(s)=\frac{\hat{k}_1 G(s)K_1(s)}{1+\hat{k}_1\hat{k}_2 K_1(s)K_2(s)G(s)}$$

用如下命令可求解其增益值，于是得 $\hat{k}_1=0.0242$ 与 $\hat{k}_2=82.7206$

```
》 k1_=dk1(2)*dg(4)/(ng(4)*nk1(2))
   k1_=
         0.0242
》 k2_=2/k1_
   k2_=
         82.7206
```

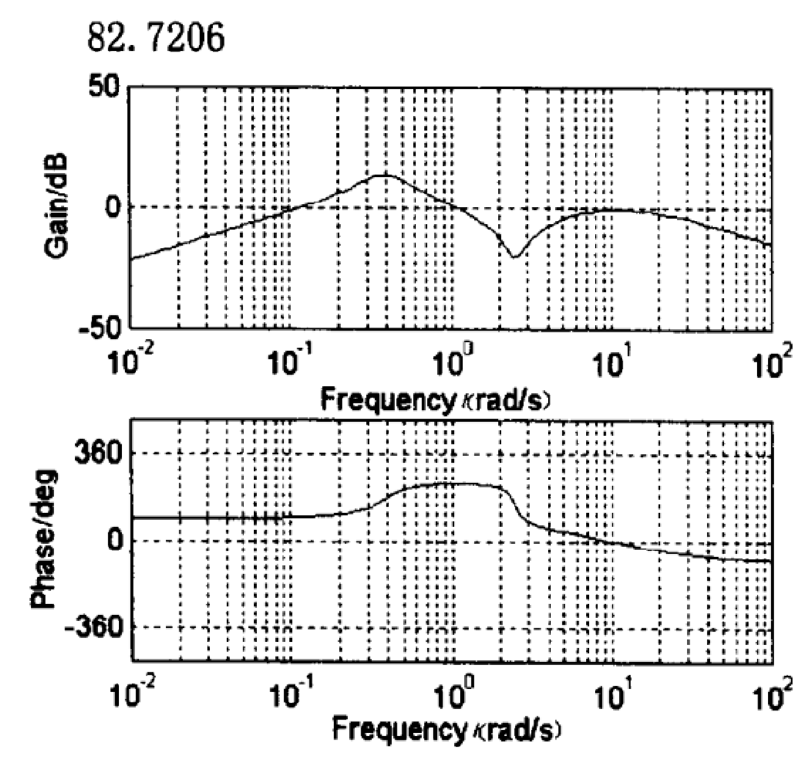


图 4-25　例 4-12 补偿后系统的伯德图

修改后的系统伯德图与系统阶跃响应见图 4-25 与图 4-26 。由图可以看出，使用该设计方法可以达到稳态输入输出比值为 1 。可以通过调整其增益分配(只要总和为 2)，来调整旋翼倾斜角度与直升飞机速度的比值。

$K_2(s)$可以近似为一个微分器。它可以由一电子器件(如运算放大器)或通过使用它装在直升飞机上测量纵向速度的加速仪表

组成。

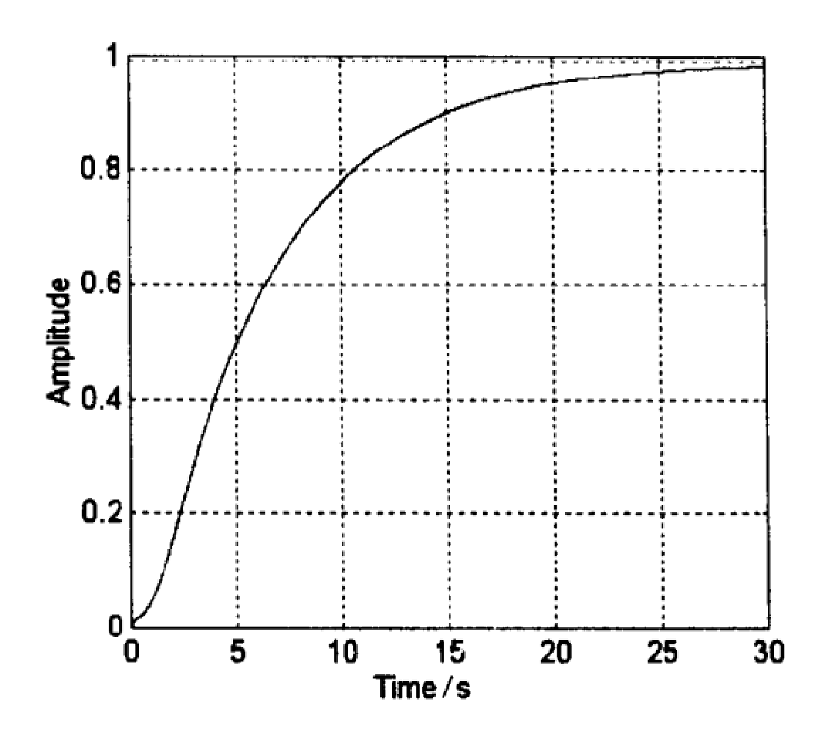


图 4-26　例 4-12 中修改增益值后的系统阶跃响应

利用 MATLAB 的 series 与 feedback 命令可求得系统的闭环传递函数；再使用 minreal 命令抵消传递函数中相同的零极点，然后利用 tf2zp 命令可求得闭环系统的零极点结果如下：

```
clzero=
        -10.0000
          0.2500+2.4975i
          0.2500-2.4975i
clpole=
        -37.1320
        -1.0325+1.1611i
        -1.0325-1.1611i
        -0.1665
  gain=
          0.2369
```

使用下列 margin 命令可以得到系统的稳定裕量的两项指标:

```
》 margin[20*log10(gm)   pm   wgc   wpc]
margin=
        -13.9762    57.4381    1.0986    0.4002
```

因为已经满足了系统稳定性的要求，也达到了阶跃输入的零稳态误差指标，所以完成了设计任务。在本章最后，将分析非最小相位系统的稳定裕量问题。

4.7 非最小相位系统的稳定裕量

正确地确定与解释非最小相位系统的稳定裕量，是一件复杂的工作。大多数书中只定义了最小相位系统的增益与相位裕量。我们将使用补偿后的直升飞机控制系统说明如何找出与解释系统的稳定裕量的方法。

由系统根轨迹图可以看出，要使系统稳定，其增益有上下限的限制。因此系统存在两个增益裕量，一个为上增益裕量 GM 和另外一个增益裕量 GRM 。由伯德图也可以看出，相频特性中存在着两个相位穿越频率，所以也存在两个增益穿越频率和两个相角裕量。 MATLAB 中的 margin 命令只能找出每个裕量中的一个，即 GM 是-14 ，而 PM 是 57°。在这个系统伯德图中存在几个附加的特点需要我们引起注意。

(1) 在伯德图中存在着几个穿越频率；

(2) 在相频穿越频率点，对应幅频特性中的增益值，在一点其增益值大于 0 ；而在另一点处增益值小于 0 。同样的现象也出现在增益穿越频率点。

(3) 尽管闭环系统是稳定的，而 GM 值是负值。由闭环控制系统稳定性分析的教课书中得知，如果系统是稳定的， GM 值一定大于零。而这种结论只适用于最小相位系统。为了能理解这一点，需要更进一步定义稳定裕量。

一个系统(如前面讨论的例题)可能会由于增益变化而使得系统不稳定，即由于增益的增加或减少而导致系统不稳定。在最小相位系统分析中， GM 表示增益可以增加的最大量(即增益的

上限值)，如超出了这个量，则系统会变为不稳定系统。而 GRM 表示增益的下限值(即增益减小的最大量)。因为该下限值小于通常的设定值，取对数以后 GRM 为负值。 margin 命令可以求得 GRM 实际值。

系统的部分频率响应参数如下所示：

	Freq	dB	Phase
	0.1000	−1.1833	98.0366
wgc1	0.1293	1.3661	100.9193
wpc1	0.3929	14.1172	176.8220
	0.4222	13.7436	189.3819
wgc2	1.0960	0.0415	237.5153
	1.1253	− 0.3476	237.3798
	2.3263	−18.4412	185.2157
wpc2	2.3556	−18.9708	180.1352
	2.3848	−19.4528	174.5451

由这些频率响应参数的变化，可以得出关于增益裕量的结论如下：

(1) 在ω = 0.39rad/s 点， GRM=−14dB 。由伯德图也可清楚地看到这一点。注意到，如果该频率点的增益减小 14dB(而相位不变)，则幅值图将下移，并且在相角为 180°的点处穿越 0dB 线。

(2) 在ω =2.35rad/s 点， GM=18.9dB 。如果该频率点的增益增加 18.9dB(而相位不变)，则幅值图将上移，并且在相角为 180°的点处穿越 0dB 线。

由奈魁斯特图(见图 4-27a)可以看出，开环传递函数具有 2 个右半平面极点。依照奈魁斯特稳定判据，因为其轨迹按顺时针方向包围(−1, j0)点两次，所以系统是稳定的。我们知道伯德图法不能够确定非最小相位闭环系统的稳定性。由图也可以看出，如果增益减小，则奈魁斯特轨迹将向右移。轨迹将不能包围(−1, j0)点。因此，系统变为不稳定。同样，如果提高增益，系统也会失去稳定性。

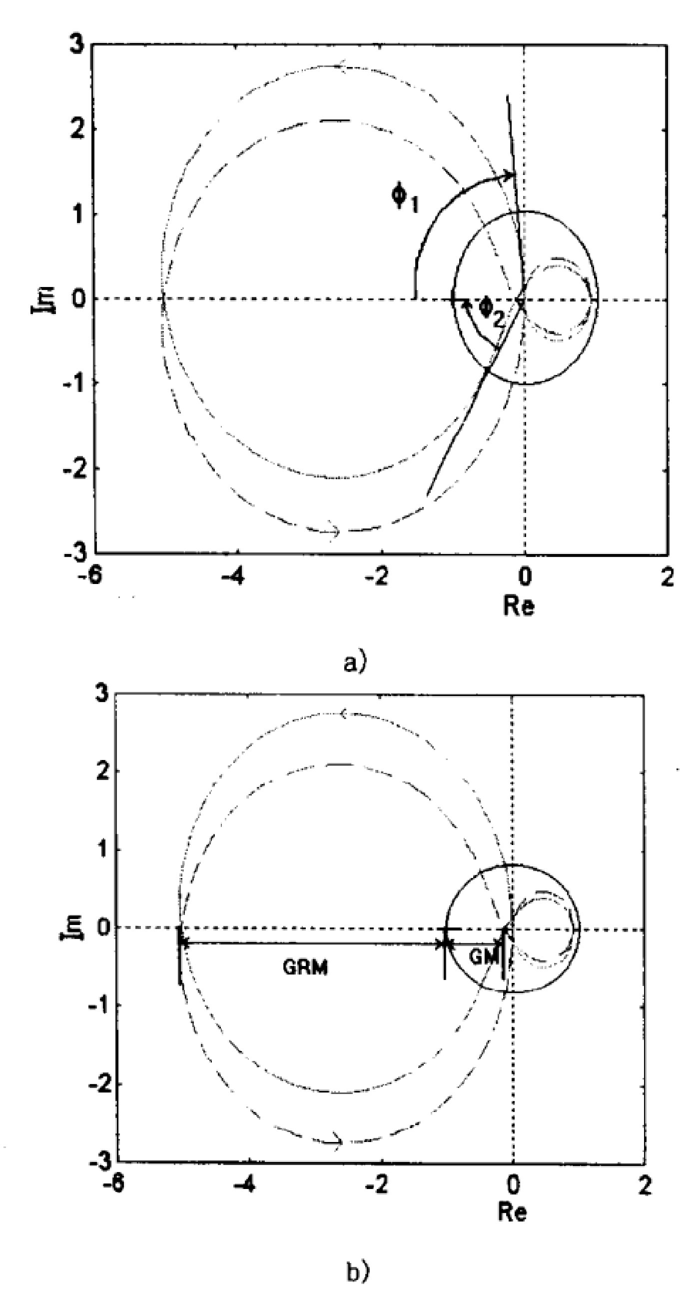


图 4-27 带有单位圆的直升飞机补偿系统的奈魁斯特图

a) 相角裕量 相角裕量（超前）ϕ_1=-80°

相角裕量（滞后）ϕ_2=57°

b) 增益裕量 上限增益裕量 GM=19dB

下限增益裕量 GRM=-14dB

相位的变化也会使系统不稳定。因此，在系统中增加负相位(或由延迟环节或极点引起的相位滞后)也会引起系统不稳定。这一点是由相角裕量 PM 衡量的。在这个例子中，ω=1.096 点处的

相角裕量是(237−180)57°。使用 margin 命令可以确定这个量。然而，添加正相角(超前相位)也会引起系统不稳定。这与通常概念不同。因为一般地讲，加入超前相位会使系统稳定裕量增加。该数据表明，在ω=0.129 点处有一个−79°的相位，相位图将向上移动，而且在增益为单位值的那点处穿越 180°线。在奈魁斯特图中，同样的变化将旋转这个图，并且使它穿越(−1,j0)点。两个相位裕量表示在图 4-27b 中。

稳定裕量其实是表明被设计的系统可以允许模型不确定性的程度。必须按频率响应数据进行计算，进一步验证并且通过根轨迹和奈魁斯特图来验证系统的稳定裕量，才能对系统这种允许模型的不确定性的程度下结论。

习　　题

4-1　试推导在 4.4.2 节中给定的根轨迹几何方法中，极点与零点位置的关系表达式。

4-2　试推导在 4.4.3 节中给定的根轨迹解析法中，τ_z与τ_p之间的关系表达式。

4-3　试推导在 4.4.4 节中给定的ω_{max}、 $\sin\phi$和 M 之间的关系表达式。

4-4　试推导在 4.4.5 节中给定的伯德图解析法中τ_z与τ_p之间的关系表达式。

4-5　下列被控传递函数都分别给出了对应的特性指标要求。请分别设计串联补偿器，使系统满足对应的特性指标要求。在设计过程中，请使用超前、滞后、 PID 这些基本环节的补偿器，或者使用与这些基本环节结合的补偿器。如果这些补偿器不能满足特性指标要求，请使用其它补偿结构。

在每一种情况中，请给出你的设计结果：补偿器传递函数、闭环传递函数、闭环零极点根轨迹、开环伯德图幅频与相频特性图、闭环伯德图幅频图、闭环阶跃响应、$\sigma\%$、 T_r、 T_s、 M_r、 Bw、 PM 和 GM 。

补偿器的设计必须符合实际情况的要求，即如果使用 PD 或 PID 调节器，为满足实际系统的要求，或者在高频处添加一个极点，或者在反馈通道中加入一微分环节。如果使用超前或滞后补偿器，要保证补偿器极点与

零点的比值小于 20 。如果这个比值必须大于 20 才能满足补偿要求，则要将该补偿器分级实现。原因在于较大的比值会使在实际补偿器电气元件选择方面遇到困难。

注：在下列问题中，采用 $e_{ss_/}$符号表示单位斜坡输入的稳态误差，而符号 $e_{ss\rfloor}$表示单位阶跃输入的稳态误差。

(1) $G(s)=\dfrac{10(s+5)}{(s+15)(s^2+8s+20)}$ $e_{ss_\rfloor}\leqslant 10\%$， $\zeta\geqslant 0.707$

(2) $G(s)=\dfrac{1}{s(s+1)(s+5)}$ $\zeta\geqslant 0.707$, $T_s\leqslant 4$, $T_r\geqslant 3s$

(3) $G(s)=\dfrac{1}{s(s+8)^2}$ $e_{ss_/}\leqslant 5\%$， $\zeta\geqslant 0.707$

(4) $G(s)=\dfrac{1}{(s+1)(s+3)}$ $e_{ss_/}\leqslant 1\%$， $\sigma\%\leqslant 10\%$， $T_s\leqslant 5s$

(5) $G(s)=\dfrac{1}{s(s+2)}$ $e_{ss_/}\leqslant 1\%$， $\sigma\%\leqslant 10\%$， $T_s\leqslant 6s$

(6) $G(s)=\dfrac{100}{s^2(s+100)}$ $\sigma\%\leqslant 10\%$， $T_s\leqslant 2s$， PM=40°

(7) $G(s)=\dfrac{1}{s(s+1)(s+5)}$ $e_{ss_/}\leqslant 1\%$， PM=40°

(8) $G(s)=\dfrac{10}{s^2(s+10)}$ $\sigma\%\leqslant 15\%$， PM=45°

(9) $G(s)=\dfrac{10(s+2)}{s(s+0.2)(s+10)}$ $\sigma\%\leqslant 10\%$， $T_s\leqslant 1s$

(10) $G(s)=\dfrac{1}{s(s+0.5)}$ $e_{ss_\rfloor}\leqslant 1\%$， $\zeta\approx 0.707$， $\omega_n\approx 5$

(11) $G(s)=\dfrac{80}{s(s+4)}$ $\zeta\geqslant 0.707$， $\omega_n\approx 10$

(12) $G(s)=\dfrac{1600}{s(s+4)(s+16)}$ $e_{ss_/}\leqslant 4\%$， PM≈30°

(13) $G(s)=\dfrac{60990}{(s+58)(s+50)(s-50)}$

在$\sigma\%<5\%$的前提下,求取最快的响应与最大的增益与相角裕量。

(14) $G(s)=\dfrac{1}{s^2}$ $T_s\leqslant 4s$， $\sigma\%\leqslant 30\%$

(15) $G(s)=\dfrac{10}{s^2}$ $\qquad T_s \leqslant 4\text{s}$， $\sigma\% \leqslant 20\%$

(16) $G(s)=\dfrac{25}{s(s+25)}$ $\qquad e_{ss_l} \leqslant 1\%$, $\text{PM} \approx 45°$

(17) $G(s)=\dfrac{1}{s(s+1)(s+2)}$ $\qquad e_{ss_l} \leqslant 20\%$, $\text{PM} \approx 45°$

4-6 倒摆问题在控制理论研究中是一个普遍被采用的范例。倒摆是一个棒通过一连接轴，其底端连接在一辆可推动的小车上。控制系统的状态参数，以保持该棒的倒立稳定。正如大家所知道的，该系统是不稳定的，因为这个摆会倾斜倒下。描述该系统特性的方程是非线性方程。系统的运动方程如下：

$$\ddot{y}=\frac{1}{(m/m_1)+\sin^2\theta}\left(\frac{u}{m}+\dot{\theta}^2 l\sin\theta-g\sin\theta\cos\theta\right)$$

$$\ddot{\theta}=\frac{1}{l(m/m_1)+\sin^2\theta}\left(-\frac{u}{m_1}\cos\theta-\ddot{\theta}^2 l\cos\theta\sin\theta+\frac{m_1+m}{m_1}g\sin\theta\right)$$

式中，m 为小车的质量；m_1 为摆的质量；l 为摆的长度；g 为重力加速度；y 为小车的位置；θ 为摆的垂直角度；u 为作用在小车上的控制输入力。

因为要保持摆的倒立，首先考虑这个模型在θ=0 点附近的线性化问题。该系统的线性状态方程模型是

$$\dot{\boldsymbol{x}}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & -m_1 g/m & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & (m+m_1)g/ml & 0\end{bmatrix}\boldsymbol{x}+\begin{bmatrix}0\\ 1/m\\ 0\\ -1/ml\end{bmatrix}\boldsymbol{u}$$

其中，状态矢量是 $\boldsymbol{x}=\begin{bmatrix}y & \dot{y} & \theta & \dot{\theta}\end{bmatrix}'$。

因为所有的状态变量是直接可测的，所以有许多可能的输出矢量 $\boldsymbol{z}$。

$$\boldsymbol{z}=\begin{bmatrix}\alpha_1 & 0 & 0 & 0\\ 0 & \alpha_2 & 0 & 0\\ 0 & 0 & \alpha_3 & 0\\ 0 & 0 & 0 & \alpha_4\end{bmatrix}\boldsymbol{x}$$

其中，如果α_i 对应的状态是可测的，α_i=1；否则α_i=0。

(1) 设定如下参数值:

$$l=1\text{m}，\quad m=1\text{kg}，\quad m_1=0.1\text{kg}，\quad g=9.8\text{m/s}^2$$

通过测量与控制小车的位置(即 $z=[1\ 0\ 0\ 0]x$)就可以使该系统稳定，试求取输入 u 与输出 y 之间的传递函数。

(2) 说明该系统是不稳定的。

(3) 设计一个可使系统稳定的补偿器。求取 GM 与 PM 。

(4) 求取闭环系统的脉冲响应及脉冲响应曲线。

(5) 设计一个补偿器，要求该补偿器可减小系统的设置时间，同时尽可能地改进系统稳定裕量。

4-7 将习题 4-6 中摆的质量换为 $m_1=1\text{kg}$ ，重复习题 4-6 的设计过程。试说明这两个问题中的不同之处。

4-8 在习题 4-6 中，可以容易地测量这个摆棒的位置。在这种情况下，有两个输出，因此

$$z=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}x$$

这是一个单输入双输出系统。

(1) 求出在输入与每一输出之间的传递函数。

(2) 选取一适当的补偿方法，并设计补偿器使得两个系统输出稳定。例如，可以考虑摆的角度作为内环控制参数，而小车的位置作为外环控制参数。

(3) 求取闭环系统的脉冲响应及脉冲响应曲线。

附录 程序清单

在下列程序中，被控对象传递函数 $G(s)$用 ng 、 dg 表示；补偿器 $K(s)$传递函数由 nk 、 dk 表示。希望的相角与增益穿越频率由 dpm 与 wgc 表示。在每段程序开始还定义了其它相关的符号。

1.Ziegler-Nichols 程序。该程序需要预先确定被控系统传递函数 ng 与 dg ； keyboard 命令可停止程序运行，以便使用 rlocfind 命令求得和定义 km 和 wm ，最后执行程序。其 MATLAB 程序清单如下:

```
ng=400； dg=[1 30 200 0]； clg
rlocus(1,[1 30 200 0])；
[km, pole]=rlocfind(ng,dg)
keyboard
wm=imag(pole(2))；
kp=0.6*km； kd=kp*pi/(4*wm)； ki=kp*wm/pi；
nk=[kp kd ki]， dk=[1 0]，
```

2.解析的PID/PD程序。该程序需要预先确定ng、dg、wgc、dpm和ki等几个参数。其MATLAB程序清单如下：

```
function [kp,kd,nk,dk]=pid(ng,dg,ki,dpm,wgc)
ngv=polyval(ng,j*wgc)； dgv=polyval(dg,j*wgc)；
g=ngv/dgv；
thetar=(dpm-180)*pi/180；
ejtheta=cos(thetar)+j*sin(thetar)；
eqn=(ejtheta/g)+j*(ki/wgc)
x=imag(eqn)；
r=real(eqn)；
kp=r
kd=x/wgc
if ki~=0，
 dk=[1 0]； nk=[kd kp ki]；
 else dk=1； nk=[kd kp]；
end；
```

3.根轨迹超前补偿的几何设计法。其MATLAB程序清单如下：

```
function [nk,dk,kc]=rllead(ng,dg,s_1)
ngv=polyval(ng,s_1)； dgv=polyval(dg,s_1)；
g=ngv/dgv；
theta=angle(g)；
```

```
if theta > 0; phi_c=pi-theta; end;
if theta < 0; phi_c=-theta; end;
phi=angle(s_1);
theta_z=(phi+phi_c)/2;
theta_p=(phi-phi_c)/2;
z_c=real(s_1)-imag(s_1)/tan(theta_z);
p_c=real(s_1)-imag(s_1)/tan(theta_p);
nk=[1 -z_c];
dk=[1 -p_c];
nkv=polyval(nk, s_1); dkv=polyval(dk, s_1);
kv=nkv/dkv;
kc=abs(1/(g*kv));
if theta < 0; k= -k; end;
```

4.根轨迹超前补偿的解析设计法。其 MATLAB 程序清单如下：

```
function [nk, dk]=anrllead(ng, dg, s_1, kc)
ngv=polyval(ng, s_1); dgv=polyval(dg, s_1);
g=ngv/dgv;
thetag=angle(g); thetag_d=thetag*180/pi;
mg=abs(g); ms=abs(s_1);
thetas=angle(s_1); thetas_d=thetas*180/pi;
tz=(sin(thetas)-kc*mg*sin(thetag-thetas))
/(kc*mg*ms*sin(thetag));
tp= -(kc*mg*sin(thetas)+sin(thetag+thetas))
/(ms*sin(thetag));
nk=[tz, 1];
dk=[tp, 1];
```

5.伯德图超前补偿的解析设计法。其 MATLAB 程序清单如下:

```
function [nk, dk]=bodelead(ng, dg, kc, w, dpm)
```

```
[mu, pu]=bode(kc*ng, dg, w);
smo=length(mu);
phi=dpm*pi/180;
a=(1+sin(phi))/(1-sin(phi));
mu_db=20*log10(mu);  mm=-10log10(a);
wgc=spline(mu_db, w, mm);
T=1/(wgc*sqrt(a));
z=a*T;  p=T;
nk=[z,  1];
dk=[p,  1];
```

6.伯德图超前/滞后补偿的解析设计法。其 MATLAB 程序清单如下:

```
function  [nk, dk]=anbdlead(ng, dg, kc, wgc, dpm)
ngv=polyval(ng, j*wgc);dgv=polyval(dg, j*wgc);
g=ngv/dgv;
thetag=angle(g);  thetag_d=thetag*180/pi
mg=abs(g)
dpm_rad=dpm*pi/180;
t_z=(1+kc*mg*cos(dpm_rad-thetag))/(-
wgc*kc*mg*sin(dpm_rad-thetag))
t_p=(cos(dpm_rad-thetag)+kc*mg)/wgc*sin(dpm_rad-
thetag))
nk=[t_z, 1]
dk=[t_p, 1]
```

7.伯德图滞后补偿设计法。其中,输入参数 w(频率矢量)必须是列矢量。其 MATLAB 程序清单如下:

```
function  [nk, dk]=bodelag(ng, dg, w, kc, dpm)
wgc=spline(pu, w, dpm-180)
mu_db=20*log10(mu);
ind=find(w=wgc);  ind=ind(1);
```

```
gr=-mu_db(ind)
alpha=10^(gr/20)
T=10/(alpha*wgc);
nk=alpha*T, 1]
dk=[T, 1]
```



第5章　调节系统的状态空间设计方法

5.1　概述

控制系统分为调节系统和跟随系统。调节系统是指系统在干扰或参数变化等情况下，保持输出不变，例如温度控制系统是维持系统温度和设定温度值一致；电压调节器是保持输出电压不变；卫星控制系统是保持发射器和地面接收天线的角度不变。以上介绍的调节系统，其设计的主要依据是期望的动态性能指标。跟随系统要求输出以较小的误差跟随输入的变化，例如自动飞行器随着运动的飞机或船舶的变化而变化。跟随系统的动态响应和稳态误差必须符合要求的性能指标。

在本章中，只讨论调节系统的设计方法，因此，着重介绍改善系统动态性能指标的方法。由于系统的动态响应主要是由它的极点位置决定的，因此调节系统的设计是应用补偿器或状态反馈将原系统的开环极点变成期望的系统极点，从而达到改善系统性能的目的。

前面对经典的根轨迹设计方法和伯德图设计方法已进行了较详细的介绍，在本章中，将介绍状态空间设计方法（亦称为极点配置方法），该方法可以得到期望的闭环极点。

5.2　极点配置方法

讨论简单的一阶方程

$$\dot{x} = x + u \qquad x(0) = x_0$$
$$y = x$$

式中，u 为系统的实际输入。在调节器系统中，外部控制输入通常置为0。上述开环系统是不稳定的。在状态空间系统设计中，

通常将反馈控制设定为

$$u = -kx$$

因此

$$\dot{x} = x + u = x - kx = (1-k)x$$

闭环极点是 $1-k$，当 $k>1$ 时，系统是渐近稳定的。实际上，适当地选择 k，可以将闭环极点放在实轴的任意位置上。

对于二阶开环系统

$$\dot{x} = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$|sI - A| = \begin{vmatrix} s-3 & 2 \\ -1 & s \end{vmatrix} = (s-1)(s-2) = 0$$

式中，I 为单位矩阵。

系统的极点是 $s_1=1$、$s_2=2$，因此，系统是不稳定的。对二阶系统能否用状态反馈将闭环极点放在左半平面，使系统稳定呢？假设系统的期望闭环极点是 p_1、p_2，因此系统的特征方程是

$$(s-p_1)(s-p_2) = s^2 + (p_1+p_2)s + p_1p_2 = s^2 + \alpha s + \beta = 0$$

假如采用状态反馈

$$u = -Kx = -[k_1 \quad k_2]x$$

即
$$u = -k_1x_1 - k_2x_2$$

对前面所述的不稳定系统，其闭环系统方程是

$$\dot{x} = \begin{bmatrix} 3 & -2 \\ 1 & 0 \end{bmatrix} x - \begin{bmatrix} 1 \\ 0 \end{bmatrix} [k_1 \quad k_2] x$$

$$= \begin{bmatrix} 3-k_1 & -2-k_2 \\ 1 & 0 \end{bmatrix} x$$

其特征方程为

$$|sI - \hat{A}| = \begin{vmatrix} s-3+k_1 & 2+k_2 \\ -1 & s \end{vmatrix} = s^2 + (k_1-3)s + (2+k_2)$$

将期望系统的特征方程和加入状态反馈的系统特征方程进

行比较，于是有

$$k_1 = \alpha + 3 \qquad k_2 = \beta - 2$$

因此，对于二阶系统，也可以得到任意的闭环极点位置。

人们会问，这种反馈控制是如何起作用和为什么会起作用？它对所有的系统都有效还是有什么条件？求取反馈控制的一般公式是什么？方法是否实用？下面对上述问题进行讨论和分析。

系统状态的定义是完全描述系统所必需的最小一组变量。也就是说，假如状态的初始值 $\boldsymbol{x}(t_0)$ 和 $t > t_0$ 的系统的输入是已知的，则可以预测系统将来的状态，这组变量，就是系统的状态变量。

按照以上定义可以看到，系统的状态反馈是非常重要和有效的。系统是用所有的必要的信息即状态来校正自己。而经典补偿器只是用输出进行反馈补偿。当然，这仅仅是一种直觉的论据，后面将更充分地证明这一点。

当设置两个特征方程的系数相等时，可以得到带未知数的一些方程，这些方程并不总是有解的。状态反馈的实用性是很微妙的和复杂的，在下面将详细讨论。

考虑 n 阶系统有

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$$

$$\boldsymbol{G}(\mathrm{s}) = \boldsymbol{C}(\mathrm{s}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}$$

当给系统一个适当的输入，系统的状态可以在状态空间内任何方向移动，即可以在复平面内任意改变极点的位置，也就是改变系统的性能，则称该系统是能控的。系统的能控性可以用下述方法检验。

对 n 阶系统，当且仅当系统的能控矩阵 $\boldsymbol{M}$ 的秩是 n 时，系统是能控的。能控矩阵 $\boldsymbol{M}$ 为

$$\boldsymbol{M} = \left[\boldsymbol{B} \vdots \boldsymbol{A}\boldsymbol{B} \vdots \boldsymbol{A}^2\boldsymbol{B} \vdots \boldsymbol{A}^{n-1}\boldsymbol{B}\right]$$

前面提到的适当的输入是指状态反馈 $\boldsymbol{u}=-\boldsymbol{K}\boldsymbol{x}$ ，因此，假如系统是能控的，则可以采用状态反馈将极点置于复平面的任意位

置。

实际上，有比能控性较弱的一个概念，即可稳定化能力，可满足大部分要求。它指的是仅配置系统不稳定模型的能力。因此，假如不稳定的模型是可控的，或着说，不可控的模型是稳定的，则称系统是可稳定的。以前检查系统可稳定性的方法是转换系统模型，检查所有特征值和在矩阵 $\boldsymbol{B}$ 中对应的行。下面是一个简单的例子。

考虑如下系统

$$\dot{x}=\begin{bmatrix}3 & 0\\0 & -1\end{bmatrix}x+\begin{bmatrix}1\\0\end{bmatrix}\boldsymbol{u} \qquad \dot{z}=\begin{bmatrix}3 & 0\\0 & 1\end{bmatrix}z+\begin{bmatrix}1\\0\end{bmatrix}\boldsymbol{u}$$

上述系统都是不可控的，因为矩阵 $\boldsymbol{M}$ 的秩是 1 。在第一个系统中，稳定模型{-1}是不可控的，而不稳定模型{3}是可控的，因此系统是可稳定的，即使用状态反馈控制 $\boldsymbol{u}=-\boldsymbol{K}x_1$，当 $\boldsymbol{K}>3$ 时，系统是稳定的。在第二个系统中，不稳定的模型{1}是不可控的，因此系统是不可稳定的。稳定性和可控性都是指系统是否能够稳定的。

另一个重要的事实是，从系统的传递函数分析中可以看到，不完全能控的系统传递函数中，有零极点对消的现象。因此，不能控系统的传递函数是不重要的。上述说明的反定理不成立，即系统的传递函数中有零极点对消的系统并不一定是不能控的。

有一些计算状态反馈增益的公式，其中比较常用的是 Ackermann 公式。对于所给的期望特征方程是

$$\Phi(s)=s^n+\alpha_1 s^{n-1}+\alpha_2 s^{n-2}+\cdots+\alpha_n$$

$$\boldsymbol{K}=[0\ 0\ 0\ \cdots\ 0\ 1]\boldsymbol{C}^{-1}\Phi(\boldsymbol{A})$$

式中

$$\Phi(\boldsymbol{A})=\boldsymbol{A}^n+\alpha_1\boldsymbol{A}^{n-1}+\alpha_2\boldsymbol{A}^{n-2}+\cdots+\alpha_n\boldsymbol{I}$$

期望极点位置的选取是任意的，满足要求的特征方程有各种优化方案，一般选取一对复数极点，以便满足系统动态响应的要求，而把其余极点放置在离虚轴很远的左半平面上，设计者完成极点选择后，其余的设计过程可以自动完成。

5.2.1 传递函数分析

应用经典的根轨迹和频域响应分析方法，对状态空间补偿器设计具有指导意义。这有两个原因：

(1) 通常一些问题是定义在频域内，如幅值裕量和相角裕量、频带宽度等指标。结果应在频域内检测，以便看是否满足上述指标。

(2) 获得补偿器的传递函数，对结果有更多的了解，与等效的经典设计进行比较。

为了比较，首先求取状态反馈系统的开环传递函数。系统状态空间表达式为

$$\dot{x} = Ax + Bu \qquad X(s) = \Phi(s)BU(s)$$
$$y = Cx + Du \qquad Y(s) = G(s)U(s)$$

式中

$$\Phi(s) = (sI - A)^{-1}$$
$$G(s) = C\Phi(s)B + D$$
$$U_1(s) = -KX(s) = -K\Phi(s)BU(s)$$

图 5-1 给出了零输入状态反馈系统的框图。

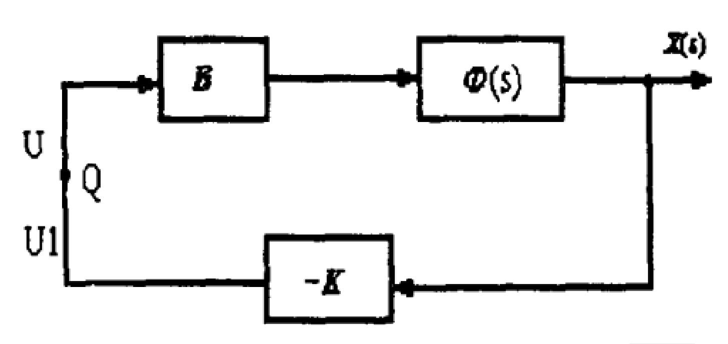


图 5-1 状态反馈系统的框图

为了计算开环传递函数，将图 5-1 所示的 Q 点打开，插入信号 $U(s)$，测量在U1处的响应 $U_1(s)$，因此开环传递函数是 $U_1(s)/U(s) = -K\Phi(s)B$。等效的经典开环传递函数是 $G(s)K(s)$。其中，$G(s)$ 是原系统的传递函数，即 $C\Phi(s)B$，$K(s)$是补偿器传递函数。开环传递函数相等，则

$$-(C\Phi B+D)K(s)=-K\Phi(s)B$$

$$K(s)=\frac{K\Phi(s)B}{C\Phi(s)B+D}$$

即假如建立起 $K(s)$，经典设计和状态空间设计的系统有着相同的动态性能，也可以应用根轨迹、伯德图和奈奎斯特分析方法分析 $K\Phi(s)B$ ，决定稳定裕量、频带宽度等。

加入给定输入 v，状态反馈系统的闭环传递函数($D=0$)是

$$u=-Kx+\overline{N}v$$

$$\dot{x}=(A-BK)x+B\overline{N}v$$

$$X(s)=(sI-A+BK)^{-1}B\overline{N}V(s)$$

$$Y(s)=C(sI-A+BK)^{-1}B\overline{N}V(s)$$

$$T(s)=C(sI-A+BK)^{-1}B\overline{N}$$

上式中，常数增益 $\overline{N}$ 是很容易计算的，对于恒定的给定输入，稳态误差决定了 $\overline{N}$。

开环传递函数和闭环传递函数的矩阵表达式为

$$KG(s)=\begin{bmatrix}A & B\\ K & 0\end{bmatrix}\qquad T(s)=\begin{bmatrix}A-BK & B\overline{N}\\ C & 0\end{bmatrix}$$

5.2.2 理论分析

下面介绍求取状态反馈增益矩阵 K 的两种方法。第一种方法是用转换矩阵 T，将状态方程转换成可控标准型，然后将期望的特征方程和加入状态反馈增益矩阵 K 后的特征方程比较，令对应项的系数相等，从而决定状态反馈增益矩阵 K。第二种方法是根据 Caylay-Hamiton 理论，它指出矩阵 $A-BK=\hat{A}$ 满足自身的特征方程，改变矩阵多项式 $\Phi(\hat{A})$的值，可以推出增益矩阵 K，由这种方法推出增益矩阵 K 的方程式叫 Ackermann 公式。下面详细地给出矩阵 K 的推导过程，并给出其 MATLAB 程序。

在下面的讨论中，假设系统是状态完全能控的。这是任意极点配置的充分必要条件。

系统方程和系统控制方程是

$$\dot{x} = Ax + Bu \tag{5-1}$$

$$u = -Kx \tag{5-2}$$

定义变换矩阵 T 为

$$T=MW \tag{5-3}$$

式中，M 为能控矩阵

$$M = \left[B \vdots AB \vdots A^2B \vdots A^{n-1}B\right] \tag{5-4}$$

W 有如下形式

$$W = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{5-5}$$

式中，a_i（$i=1,2,\cdots,n-1$）是系统特征多项式的系数。

$$|sI - A| = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_n$$

定义新的状态矢量 $\hat{x}$ 为

$$x = T\hat{x}$$

因为系统是状态完全能控的，矩阵 M 的秩是 n。由于 T 的逆矩阵存在，方程式(5-1)可以写为

$$\dot{\hat{x}} = T^{-1}AT\hat{x} + T^{-1}Bu \tag{5-6}$$

式中，$T^{-1}AT$ 可为如下形式

$$T^{-1}AT = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \tag{5-7}$$

$$T^{-1}B = [0 \quad 0 \quad 0 \quad \cdots \quad 0 \quad 1]^{\mathrm{T}} \tag{5-8}$$

选择系统的期望特征值是 μ_1、μ_2、…、μ_n。期望特征方程是

$$(s-\mu_1)(s-\mu_2)\cdots(s-\mu_n) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n \tag{5-9}$$

令 $\quad \hat{K} = KT = [\delta_n \quad \delta_{n-1} \quad \cdots \quad \delta_1]$ (5-10)

$$u = -Kx = -KT\hat{x} = -\hat{K}\hat{x} \tag{5-11}$$

当方程式(5-11)用于控制系统方程式(5-6)，系统方程变为

$$\hat{\boldsymbol{x}} = \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T}\hat{\boldsymbol{x}} - \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{K}\boldsymbol{T}\hat{\boldsymbol{x}}$$

则变换后的系统特征方程是

$$\left|s\boldsymbol{I} - \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} + \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{K}\boldsymbol{T}\right| = 0$$

系统的特征方程是相等的，即

$$\left|s\boldsymbol{I} - \boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}\right| = \left|s\boldsymbol{I} - \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} + \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{K}\boldsymbol{T}\right|$$

参照方程式(5-7)、式(5-8)和式(5-10)可得

$$\left|s\boldsymbol{I} - \boldsymbol{T}^{-1}\boldsymbol{A}\boldsymbol{T} + \boldsymbol{T}^{-1}\boldsymbol{B}\boldsymbol{K}\boldsymbol{T}\right|$$

$$= \left|s\boldsymbol{I} - \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} \delta_n & \delta_{n-1} & \cdots & \delta_1 \end{bmatrix}\right|$$

$$= \begin{vmatrix} s & -1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & -1 \\ a_n + \delta_n & a_{n-1} + \delta_{n-1} & \cdots & s + a_1 + \delta_1 \end{vmatrix}$$

$$= s^n + (a_1 + \delta_1)s^{n-1} + \cdots + (a_{n-1} + \delta_{n-1})s + (a_n + \delta_n) = 0 \qquad (5\text{-}12)$$

这是有着状态反馈的系统的特征方程，它应该和方程式(5-11)相等，因此相应项的系数相同，故有

$$a_1 + \delta_1 = \alpha_1, \qquad a_2 + \delta_2 = \alpha_2, \quad \ldots\ , \ a_n + \delta_n = \alpha_n$$

将上述结果代入方程式(5-10)可得

$$\boldsymbol{K} = \hat{\boldsymbol{K}}\boldsymbol{T}^{-1} = \begin{bmatrix} \delta_n & \delta_{n-1} & \cdots & \delta_1 \end{bmatrix}\boldsymbol{T}^{-1}$$

$$= \begin{bmatrix} \alpha_n - a_n \ \vdots \ \alpha_{n-1} - a_{n-1} \ \vdots \ \cdots \ \vdots \ \alpha_2 - a_2 \ \vdots \ \alpha_1 - a_1 \end{bmatrix}\boldsymbol{T}^{-1} \qquad (5\text{-}13)$$

因此，假如系统是状态完全能控的，选取矩阵 $\boldsymbol{K}$ 可以实现系统极点的任意配置，即可以使系统达到要求的性能指标。

根据上面的分析，设计状态反馈矩阵 $\boldsymbol{K}$，以便得到要求的极点配置的步骤如下：

(1) 检测能控矩阵。能控矩阵的秩必须是 n，否则不能应用方程式(5-13)。

(2) 从矩阵 $\boldsymbol{A}$ 得出特征多项式

$$|sI-A| = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_n$$

从而决定 a_1， a_2，…， a_n 的值。

(3) 求取变换矩阵 $\boldsymbol{T}$

$$\boldsymbol{T}=\boldsymbol{MW}$$

式中， $\boldsymbol{M}$ 和 $\boldsymbol{W}$ 由方程式(5-4)和式(5-5)给出。

(4) 由期望的特征值写出特征多项式

$$(s-\mu_1)(s-\mu_2)\cdots(s-\mu_n) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n$$

从而决定 α_1、α_2、…、α_n 的值。

(5) 求取状态反馈增益矩阵 $\boldsymbol{K}$

$$\boldsymbol{K}=[\, \alpha_n - a_n \;\vdots\; \alpha_{n-1} - a_{n-1} \;\vdots\; \cdots \;\vdots\; \alpha_2 - a_2 \;\vdots\; \alpha_1 - a_1 \,]\boldsymbol{T}^{-1}$$

下面介绍用 Ackermann 公式求取状态反馈矩阵的方法

系统方程和控制方程如方程式(5-1)和式(5-2)所示，将式(5-2)代入式(5-1)可得

$$\dot{\boldsymbol{x}} = (\boldsymbol{A}-\boldsymbol{BK})\boldsymbol{x}$$

令

$$\hat{\boldsymbol{A}} = \boldsymbol{A}-\boldsymbol{BK}$$

期望的特征方程是

$$|sI - A + BK| = |sI + \hat{A}| = (s-\mu_1)(s-\mu_2)\cdots(s-\mu_n) =$$

$$s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n = 0$$

定义 $\Phi(s) = s^n + \alpha_1 s^{n-1} + \cdots + \alpha_{n-1}s + \alpha_n$

则

$$\Phi(A) = A^n + \alpha_1 A^{n-1} + \cdots + \alpha_{n-1}A + \alpha_n I \tag{5-14}$$

因为 $\hat{A}$ 满足期望的特征方程，因此

$$\Phi(\hat{A}) = \hat{A}^n + \alpha_1 \hat{A}^{n-1} + \cdots + \alpha_{n-1}\hat{A} + \alpha_n I = 0 \tag{5-15}$$

下面利用方程式(5-15)推导 Ackermann 公式。

为简化推导过程，考虑 $n=3$ 。

$$\hat{A} = A - BK$$

$$\hat{A}^2 = A^2 - ABK - BK\hat{A}$$

$$\hat{A}^3 = A^3 - A^2BK - ABKA - BK\hat{A}^2$$

将上述方程分别乘以α_3、α_2、α_1、α_0($\alpha_0=1$)，并相加，从而得

$$\begin{aligned}
&\alpha_3 \boldsymbol{I}+\alpha_2 \hat{\boldsymbol{A}}+\alpha_1 \hat{\boldsymbol{A}}^2+\alpha_0 \hat{\boldsymbol{A}}^3 \\
&=\alpha_3 \boldsymbol{I}+\alpha_2(\boldsymbol{A}-\boldsymbol{B K})+\alpha_1\left(\boldsymbol{A}^2-\boldsymbol{A B K}-\boldsymbol{B K} \hat{\boldsymbol{A}}\right)+ \\
&\quad \boldsymbol{A}^3-\boldsymbol{A}^2 \boldsymbol{B K}-\boldsymbol{A B K} \hat{\boldsymbol{A}}-\boldsymbol{B K} \hat{\boldsymbol{A}}^2 \\
&=\alpha_3 \boldsymbol{I}+\alpha_2 \boldsymbol{A}+\alpha_1 \boldsymbol{A}^2+\boldsymbol{A}^3-\alpha_2 \boldsymbol{B K}-\alpha_1 \boldsymbol{A B K}- \\
&\quad \alpha_1 \boldsymbol{B K} \hat{\boldsymbol{A}}-\boldsymbol{A}^2 \boldsymbol{B K}-\boldsymbol{A B K} \hat{\boldsymbol{A}}-\boldsymbol{B K} \hat{\boldsymbol{A}}^2
\end{aligned} \tag{5-16}$$

参考方程式(5-14)～式(5-16)，有

$$\begin{aligned}
\Phi(\hat{\boldsymbol{A}}) &=\Phi(\boldsymbol{A})-\alpha_2 \boldsymbol{B K}-\alpha_1 \boldsymbol{B K} \hat{\boldsymbol{A}}-\boldsymbol{B K} \hat{\boldsymbol{A}}^2- \\
&\quad \alpha_1 \boldsymbol{A B K}-\boldsymbol{A B K} \hat{\boldsymbol{A}}-\boldsymbol{A}^2 \boldsymbol{B K} \\
&=\Phi(\boldsymbol{A})-\boldsymbol{B}(\alpha_2 \boldsymbol{K}+\alpha_1 \boldsymbol{K} \hat{\boldsymbol{A}}+\boldsymbol{K} \hat{\boldsymbol{A}}^2)- \\
&\quad \boldsymbol{A B}(\alpha_1 \boldsymbol{K}+\boldsymbol{K} \hat{\boldsymbol{A}})-\boldsymbol{A}^2 \boldsymbol{B K}
\end{aligned}$$

因为　$\Phi(\hat{\boldsymbol{A}})=0$，因此有：

$$\begin{aligned}
\Phi(\boldsymbol{A}) &=\boldsymbol{B}(\alpha_2 \boldsymbol{K}+\alpha_1 \boldsymbol{K} \hat{\boldsymbol{A}}+\boldsymbol{K} \hat{\boldsymbol{A}}^2)+ \\
&\quad \boldsymbol{A B}(\alpha_1 \boldsymbol{K}+\boldsymbol{K} \hat{\boldsymbol{A}})-\boldsymbol{A}^2 \boldsymbol{B K} \\
&=[\boldsymbol{B} \vdots \boldsymbol{A B} \vdots \boldsymbol{A}^2 \boldsymbol{B}]\begin{bmatrix}
\alpha_2 \boldsymbol{K}+\alpha_1 \boldsymbol{K} \hat{\boldsymbol{A}}+\boldsymbol{K} \hat{\boldsymbol{A}}^2 \\
\alpha_1 \boldsymbol{K}+\boldsymbol{K} \hat{\boldsymbol{A}} \\
\boldsymbol{K}
\end{bmatrix}
\end{aligned} \tag{5-17}$$

因为系统是状态完全能控的，能控矩阵 $\boldsymbol{M}$ 的逆矩阵存在，对方程式(5-17)两边左乘 $\boldsymbol{M}^{-1}$，再左乘$[0\quad 0\quad 1]$，可得

$$\begin{gathered}
[0\quad 0\quad 1]\left[\boldsymbol{B} \vdots \boldsymbol{A B} \vdots \boldsymbol{A}^2 \boldsymbol{B}\right]^{-1} \Phi(\boldsymbol{A})= \\
[0\quad 0\quad 1]\begin{bmatrix}
\alpha_2 \boldsymbol{K}+\alpha_1 \boldsymbol{K} \hat{\boldsymbol{A}}+\boldsymbol{K} \hat{\boldsymbol{A}}^2 \\
\alpha_1 \boldsymbol{K}+\boldsymbol{K} \hat{\boldsymbol{A}} \\
\boldsymbol{K}
\end{bmatrix}=\boldsymbol{K}
\end{gathered}$$

因此，状态反馈增益矩阵 $\boldsymbol{K}$ 为

$$\boldsymbol{K}=[0\quad 0\quad 1]\left[\boldsymbol{B} \vdots \boldsymbol{A B} \vdots \boldsymbol{A}^2 \boldsymbol{B}\right]^{-1} \Phi(\boldsymbol{A})$$

其通用性公式为

$$K=\begin{bmatrix}0 & 0 & \cdots & 0 & 1\end{bmatrix}\begin{bmatrix}B \vdots AB \vdots \cdots \vdots A^{n-1}B\end{bmatrix}^{-1}\Phi(A)$$
$$=\begin{bmatrix}0 & 0 & \cdots & 0 & 1\end{bmatrix}M^{-1}\Phi(A) \tag{5-18}$$

方程式(5-18)称为 Ackermann 公式。

【例 5-1 】调节系统方程如下：

$$\frac{Y(s)}{U(s)}=\frac{10}{(s+1)(s+2)(s+3)}$$

定义状态变量为

$$x_1 = y$$
$$x_2 = \dot{x}_1$$
$$x_3 = \dot{x}_2$$

采用状态反馈控制 $u=-Kx$，闭环系统的期望极点为

$$\mu_1=-1+\mathrm{j}\,2\sqrt{3} \qquad \mu_2=-1-\mathrm{j}\,2\sqrt{3} \qquad \mu_3=-10$$

求取状态反馈增益矩阵 K。

解 系统的状态空间方程是

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -6 & -11 & -6\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}0\\ 0\\ 10\end{bmatrix}u$$

$$y=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix}+[0]u$$

因此

$$A=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -6 & -11 & -6\end{bmatrix} \qquad B=\begin{bmatrix}0\\ 0\\ 10\end{bmatrix}$$
$$C=\begin{bmatrix}1 & 0 & 0\end{bmatrix} \qquad D=[0]$$

能控矩阵 M 为

$$M=[B \vdots AB \vdots A^2B]$$

M 的秩为 3，因此极点配置是可行的。

下面求取特征方程的系数 a_1、a_2、a_3，可以用如下的命令求取 p=poly(A)，a_1=6、a_2=11、a_3=6。

定义矩阵 W 为

$$\boldsymbol{W}=\begin{bmatrix} a_2 & a_1 & 1 \\ a_1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}=\begin{bmatrix} 11 & 6 & 1 \\ 6 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

求取转换矩阵 $\boldsymbol{T}=\boldsymbol{MW}$。

求取期望特征方程。定义矩阵 J 为

$$\boldsymbol{J}=\begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix}=\begin{bmatrix} -2+\mathrm{j}2\sqrt{3} & 0 & 0 \\ 0 & -2-\mathrm{j}2\sqrt{3} & 0 \\ 0 & 0 & -10 \end{bmatrix}$$

用 poly(J)命令可以求取 α_1、α_2、α_3 为

$\alpha_1=14$　　$\alpha_2=56$　　$\alpha_3=160$

则要求的状态反馈增益矩阵 K 为

$$\boldsymbol{K}=[\alpha_3-a_3,\alpha_2-a_2,\alpha_1-a_1]\boldsymbol{T}^{-1}$$

MATLAB 程序 5-1 是解上述矩阵 K 的程序清单。

```
% MATLAB PROGRAM 5-1
% Pole placement
A=[0 1 0;0 0 1;-6 -11 -6];
B=[0;0;10];
JA=poly(A);
a1=JA(2);a2=JA(3);a3=JA(4);
M=[B A*B A^2*B];
rank(M)
W=[a2 a1 1;a1 1 0;1 0 0];
T=M*W;
J=[-2+j*2*sqrt(3) 0 0;0 -2-j*2*sqrt(3) 0;0 0 -10];
JJ=poly(J)
```

```
aa1=JJ(2);aa2=JJ(3);aa3=JJ(4);
K=[aa3-a3 aa2-a2 aa1-a1]*(inv(T))
end
```

下面用 Ackermann 公式求状态反馈矩阵 $\boldsymbol{K}$，首先计算矩阵的特征多项式 $\boldsymbol{\Phi}(\boldsymbol{A})$， MATLAB 的 polyvalm 命令可完成这种功能，如

$$\Phi(\boldsymbol{A}) = \boldsymbol{A}^3 + \alpha_1\boldsymbol{A}^2 + \alpha_2\boldsymbol{A} + \alpha_3\boldsymbol{I}$$

对于期望的极点，对应的特征多项式可用如下命令求取：

```
J=[-2+j*2*sqrt(3) 0 0;0 -2-j*2*sqrt(3) 0;0 0 -10]
poly(J)
poly(J)=
    1.0000    14.0000    56.0000   160.0000
```

在 MATLAB 程序中，可以用如下命令计算 $\boldsymbol{\Phi}(\boldsymbol{A})$：

```
A=[0 1 0;0 0 1;-6 -11 -6];
J=[-2+j*2*sqrt(3) 0 0;0 -2-j*2*sqrt(3) 0;0 0 -10];
JJ=poly(J);
Phi=polyvalm(poly(J),A)
Phi =
     154.0000    45.0000     8.0000
     -48.0000    66.0000    -3.0000
      18.0000   -15.0000    84.0000
```

MATLAB 程序 5-2 是用来求取状态反馈增益矩阵 $\boldsymbol{K}$ 的程序清单。

```
% MATLAB PROGRAM 5-2
% Pole placement  use of Ackermann's formula
A=[0 1 0;0 0 1;-6 -11 -6];
B=[0;0;10];
M=[B A*B A^2*B];
rank(M);
```

```
J=[-2+j*2*sqrt(3)  0  0; 0  -2-j*2*sqrt(3)  0;
   0  0  -10];
JJ=poly(J);
Phi=polyvalm(poly(J),A)
K=[0 0 1]*(inv(M))*Phi
end
```

由上述程序可求得状态反馈增益矩阵 $\boldsymbol{K}$ 为

$$\boldsymbol{K}=\begin{bmatrix}15.4 & 4.5 & 0.8\end{bmatrix}$$

5.3 用于状态空间设计的 MATLAB 命令

在极点配置和观测器设计中常用的 MATLAB 命令有 ctrb 、obsv 、 place 、 augstate 、 feedback 、 series 等。下面简单地介绍这些命令的用法及作用。详细的请参阅有关手册。

假设所给系统是

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}$$
$$\boldsymbol{y}=\boldsymbol{C}\boldsymbol{x}+\boldsymbol{D}\boldsymbol{u}$$

1. ctrb 命令　ctrb 命令用于求取系统的能控矩阵 $\boldsymbol{M}$ ，判断系统的能控性。命令的格式为

M=ctrb(A,B)

式中， $\boldsymbol{M}=[\boldsymbol{B}\vdots\boldsymbol{AB}\vdots\boldsymbol{A}^2\boldsymbol{B}\vdots\boldsymbol{A}^{n-1}\boldsymbol{B}]$ ， rank(M)可以得到能控矩阵 $\boldsymbol{M}$ 的秩，若 $\boldsymbol{M}$ 的秩是 n ，则系统是状态完全能控的，可以实现极点的任意配置。

2. obsv 命令　obsv 命令用于求取系统的能观性矩阵 $\boldsymbol{N}$ ，判断系统的状态能观测性。命令格式为

N=obsv(A,C)

该命令用于系统观测器设计。

3. place 命令　对于所给的系统方程和期望的闭环系统极点，采用 place 命令可以直接求取状态反馈增益矩阵 $\boldsymbol{K}$ 。假设系统的期望闭环极点矢量为 $\boldsymbol{p}$ ，则用以下命令可求取 $\boldsymbol{K}$ ：

K=place(A,B,p)

对于例 5-1，采用如下程序，可实现期望极点配置，并求取状态反馈矩阵 $\boldsymbol{K}$，使前面的程序大大简化。

```
% Pole placement
A=[0 1 0;0 0 1;-6 -11 -6];B=[0;0;10];
p=[-2+j*2*sqrt(3) -2-j*2*sqrt(3) -10];
K=place(A,B,p)
end
```

4. augstate 命令　该命令是将状态变量扩展为输出，为后面的状态反馈作准备。该命令格式为

[Aa Ba Ca Da]=augstate(A,B,C,D)

该命令将原系统方程变换成如下的状态方程和输出方程：

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{x} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{I} \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{0} \end{bmatrix}\boldsymbol{u}$$

命令中参数与上式的对应关系为

$$\mathrm{Aa} = \boldsymbol{A} \quad \mathrm{Ba} = \boldsymbol{B} \quad \mathrm{Ca} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{I} \end{bmatrix} \quad \mathrm{Da} = \begin{bmatrix} \boldsymbol{D} \\ \boldsymbol{0} \end{bmatrix}$$

将 augstate 、 feedback 、 series 命令结合起来，可以获得开环系统和闭环系统矩阵。

【例 5-2 】状态反馈设计。

系统状态空间方程是

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -1.4 & 9.8 & -0.02 \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 6.3 \\ 0 \\ 9.8 \end{bmatrix}\boldsymbol{u}$$

$$\boldsymbol{y} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\boldsymbol{x}$$

系统的期望极点是-1+j 、-1-j 、-2，求系统的状态反馈矩阵 $\boldsymbol{K}$。

解　首先分析一下系统的稳定性。令

$$A=\begin{bmatrix}-0.4 & 0 & -0.01\\ 1 & 0 & 0\\ -1.4 & 9.8 & -0.02\end{bmatrix}\quad B=\begin{bmatrix}6.3\\ 0\\ 9.8\end{bmatrix}$$

$$C=\begin{bmatrix}0 & 0 & 1\end{bmatrix}\qquad D=0$$

用如下命令：

```
[zT pT gainT]=ss2zp(A,B,C,D)
```

可以求取系统的零、极点和直流增益，从而可判断系统的稳定性，其结果为

```
zT=
  0.2500+2.4975j
  0.2500-2.4975j
pT=
  -0.6565
  0.1183+0.3678j
  0.1183-0.3678j
gainT=
    9.8000
```

其中， zT 为零点， pT 为极点， gainT 为增益。

从上面的结果可知，系统在右半平面有一对复数极点，因此系统是不稳定的。采用状态反馈，使系统的期望极点在

```
p=[-2 -1+j -1-j]
```

应用 place 命令可以求出状态反馈增益矩阵 $\boldsymbol{K}$ 。

```
K=place(A,B,p)
```

结果是

```
K=[0.4706 1.000 0.0627]
```

下面对加入状态反馈后系统的性能进行分析和讨论。

首先分析一下根轨迹图。可用如下命令求取

```
agk=A;  bgk=B;  cgk=C;  dgk=0;
rlocus(agk, bgk, cgk, dgk)
```

根轨迹图见图 5-2，从根轨迹上可见，在较高增益时，系统

总是稳定的。

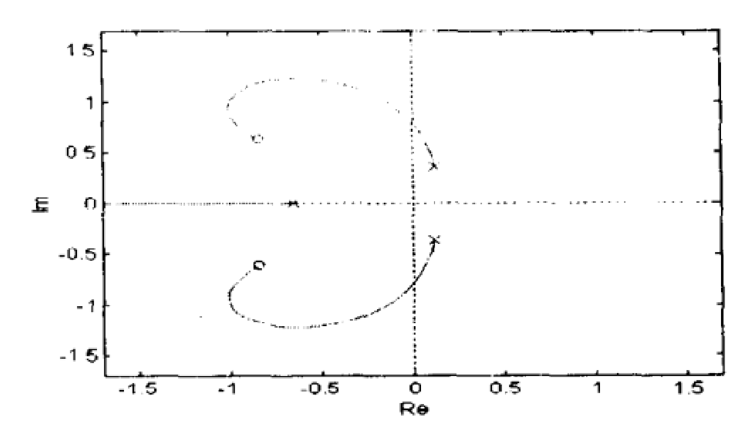


图 5-2　加入状态反馈后系统的根轨迹

可用如下命令求取加入状态反馈后系统的伯德图。

[mc pc]=bode(agk,bgk,cgk,dgk,1,ω)

系统的伯德图见图 5-3 。

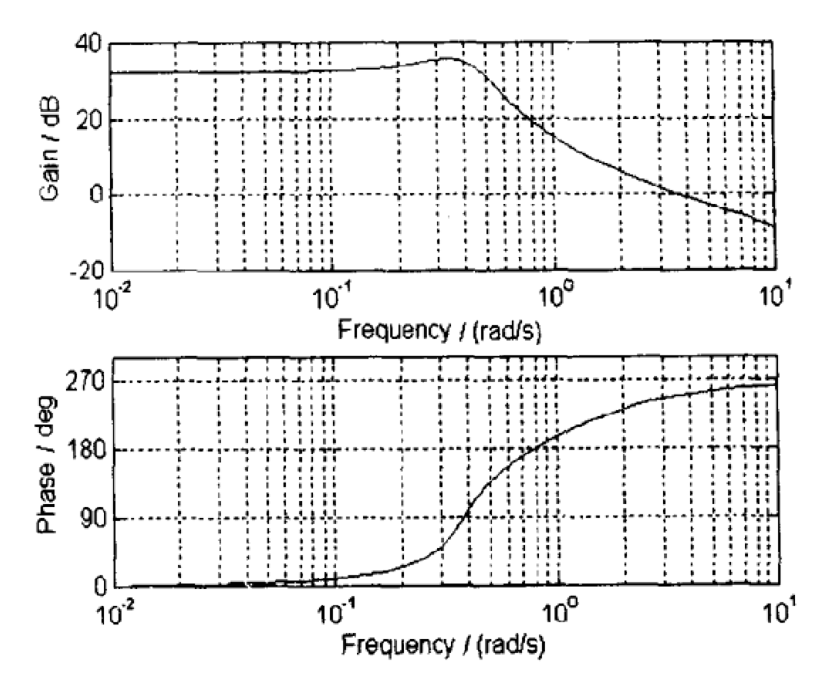


图 5-3　加入状态反馈后系统的伯德图

从伯德图可见，系统的增益裕量是无穷大。

采用如下程序求取加入状态反馈后系统的零极点位置：

```
At=A-B*K;  Bt=B;  Ct=C;  Dt=D;
[zT pT, gainT]=ss2zp(At,Bt,Ct,Dt)
```

系统的零极点为

```
zT=
   0.2500+2.4975j
   0.2500-2.4975j
pT=
   -2.0000
   -1.0000+1.0000j
   -1.0000-1.0000j
gainT=
      9.8000
```

闭环极点在期望的位置。零点没有改变。因此，状态反馈仅能改变极点位置，而无法改变零点位置。

其系统阶跃响应可以用如下程序求取：

```
dcg=dcgain(At,Bt,Ct,Dt);
yc=step(At,Bt,Ct,Dt,1,t);yc1=yc/dcg;
plot(t, yc1)
```

系统的阶跃响应见图 5-4 。

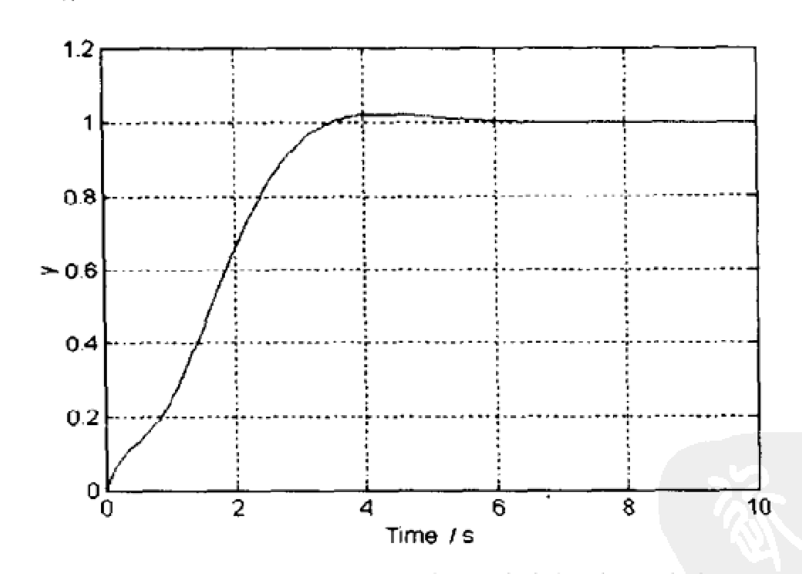


图 5-4　加入状态反馈后系统的阶跃响应

从阶跃响应可以看出，加入状态反馈后系统的响应是非常稳定的，超调量很小，过渡过程时间大约 4s ，假如状态是可观测的，则上述系统是可以实现的。

采用如下命令可以判断系统的能观性：

```
N=obsv(A,C);
rank(N)
ans=
    3
```

因为能观矩阵 N 的秩为 3，因此系统是完全能观的，上述系统是可实现的。反馈矩阵 K 的设计及系统性能分析程序见本章附录中的程序 1 。

5.4 观测器的设计

系统由状态反馈可以任意配置极点，从而得到期望的性能指标。在实际系统中，测量所有的状态有时是不可能或不实际的，状态反馈要求对所有的状态由理想的传感器测量，而实际的传感器有着有限的频带宽度。有时系统的状态不是实际的物理量，因此是不能测量的。即使所有的状态都是实际的物理量，测量所有的状态在技术上和经济上也存在着问题。除了测量问题，每个反馈环都需要实际硬件或计算机控制软件，增加了系统的复杂性，降低了系统的可靠性。假如一个传感器发生故障，整个系统变得不稳定了。实际上，仅仅某些状态或它们的组合，如输出状态是可测量的。仅仅由可测量的系统输出构成的反馈称之为输出反馈。

系统的输入和输出一般是可以测量的，可以用系统的输入和输出信号来估计系统的状态，这种估计系统状态的装置或模型即为观测器。令 $\hat{\boldsymbol{x}}$ 为状态变量 $\boldsymbol{x}$ 的估计，则有如下方程：

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{5-19}$$

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} \qquad \hat{\boldsymbol{x}}(0) = \hat{\boldsymbol{x}}_0 \tag{5-20}$$

令 $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$，则由方程式(5-19)和式(5-20)可得

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}\tilde{\boldsymbol{x}} \qquad \tilde{\boldsymbol{x}}(0) = \tilde{\boldsymbol{x}}_0 = \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0$$

$$\tilde{\boldsymbol{x}}(t) = \mathrm{e}^{\boldsymbol{A}t}\tilde{\boldsymbol{x}}(0)$$

因此，只要系统是稳定的，则观测器的误差趋近零。我们并不知道系统的初始条件，如果能知道系统的初始条件就可解系统

方程，并且计算系统在任何时刻的状态。因此，主要问题是估计系统的初始条件。注意上面所提到的观测器并没有利用输出信息，假如将系统输出和观测器的输出相比较，并用误差校正装置，就可以让误差的性能稳定。众所周知，反馈可以使不稳定的系统稳定化。因此，可以组成闭环观测器。

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \qquad \hat{x}(0) = \hat{x}_0$$

则误差系统的状态方程为

$$\dot{\tilde{x}} = (A - LC)\tilde{x} \qquad \tilde{x}(0) = \tilde{x}_0 \tag{5-21}$$

假如可以选择矩阵 L，使 $A\text{–}LC$ 的特征值是任意期望的值，则对于任意的初始条件，可以保证观测器的误差均趋近于零。上述的性质是可控双重性，被称作能观性。它可以由系统的输出状态估计系统状态。能观性是指选择某增益矩阵 L，能够使 $(A\text{–}LC)$ 的特征值在任意期望的位置。能观性可以用检测能观矩阵 N 的秩来判断，如果秩等于 n，则系统是能观的。能观矩阵定义为

$$N = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

能控性和能观性是双重概念，假如用 A'代替 A，用 C'代替 B 可以将能控矩阵转化为能观矩阵。可稳定性的双重概念是可测试性，如果说一个系统是可测试的，是指不稳定的模型是能观的，或不能观的模型是稳定的。

由以上的推理，可以用 place 命令解矩阵 L。

```
L=place(A',C',op)
```

式中， op 是观测器的期望极点矢量。观测器增益矩阵 L 是以行矢量的形式给出的，应转换成列矢量。应该注意，观测器极点的配置必须保证估测的快速性，不影响原系统的动态性能。而极点也不能在左半平面太远的位置，否则 K 和 L 的值会很大，会产生饱和甚至不稳定。同时观测器带宽增加，会引起噪声问题。因

此，设计者必须适当地选择极点的位置。

上面讨论了如何估测系统的状态，下面研究控制问题。用估测的状态代替实际状态构成反馈系统，这使得状态反馈变成现实可行的。然而，如何保证整个闭环控制系统的稳定呢？毕竟上面所用的是估测的状态，而不是实际的状态。解决的办法是可以将控制问题和观测问题分离，即首先假设状态是可测量的，或可观测的，按照期望的系统性能求取控制器。然后，设计系统观测器，观测系统的状态，用估测的状态构成上述系统。系统的闭环极点是控制器极点和观测器极点的组合。上述结果被称为分离特性，是现代控制理论的基础之一。分离特性可以表示为系统、控制器、观测器误差系统的结合，由下列方程式表示：

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A-BK & -BK \\ 0 & A-LC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} \tag{5-22}$$

三角矩阵的特征值是对角矩阵特征值的组合，即矩阵 $\begin{bmatrix} A-BK & -BK \\ 0 & A-LC \end{bmatrix}$ 的特征值是矩阵 $A-BK$ 和矩阵 $A-LC$ 的特征值的组合，即分离特性。

传递函数分析：有状态反馈和观测器系统的补偿器方程式是

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - Cx)$$

$$u = -K\hat{x}$$

将上式进行拉普拉斯变换，求可 $K(s)$ 得

$$K(s) = (sI - A + BK + LC)^{-1} L \tag{5-23}$$

$$K(s) = \begin{bmatrix} A-BK-LC & L \\ K & 0 \end{bmatrix} \tag{5-24}$$

即使 $A-BK$ 和 $A-LC$ 是稳定的，补偿矩阵 $A-BK-LC$ 也可能是不稳定的。从下面的例子中的根轨迹可以看到，两者的组合是一个条件稳定系统。

用 series 命令可以获得开环传递函数 $G(s)$ 和 $K(s)$，用 feedback 命令可以求取系统的闭环传递函数。

$$u = -K\hat{x} + \overline{N}v \tag{5-25}$$

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A-BK-LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B\overline{N} \\ B\overline{N} \end{bmatrix} v \tag{5-26}$$

$$T(s) = \begin{bmatrix} A & -BK & B\overline{N} \\ LC & A-BK-LC & B\overline{N} \\ C & 0 & 0 \end{bmatrix} \tag{5-27}$$

【例 5-3】 观测器设计。

在例 5-2 所设计的控制器的基础上设计观测器，观测器的极点为[-3+3j -3-3j -4]，从极点的配置上保证观测器的快速性比控制器的快，观测器不影响系统的性能。

解 用下列 place 命令可求观测器增益矩阵：

```
A=[-0.4  0  -0.01; 1  0  0; -1.4  9.8  -0.02];
B=[6.3; 0; 9.8];  C=[0  0  1];  D=0;
op=[-3+3*j  -3-3*j  -4];
L=place(A', C', op);  L=L'
L=
   5.4664
   4.6762
   9.5800
```

下面是求取系统零极点的程序清单：

```
Ak=A-B*k-LC-L*D*k;Bk=L; Ck=k;
[tmp1    tmp2]=size(D);
Dk=zero(tmp1  tmp2);
[zk  pk  gaink]=ss2zp(Ak, Bk, Ck, Dk, 1)
```

由上述程序可得

```
zk=
   -0.6416+0.3658j
   -0.6416-0.3658j
pk=
   -14.3751
```

```
    0.3975+4.0793j
    0.3975-4.0793j
gaink=
      9.8
```

所以

$$K_2(s)=\frac{9.8(s+0.64+\mathrm{j}0.36)(s+0.64-\mathrm{j}0.36)}{(s+14.37)(s-0.39+\mathrm{j}4.07)(s-0.39-\mathrm{j}4.07)}$$

将 $K_2(s)$ 与超前滞后补偿器比较，可以看到，在经典设计中，补偿器的零极点必须严格限制在负实轴上，而在状态空间设计中，补偿器的零极点没有严格的限制，可以位于复平面的任意位置，由此可以实现闭环极点的任意配置。

由下面程序求取开环传递函数，从而得到图5-5的根轨迹图、图5-6的伯德图和幅值、相角裕量。

```
[agk bgk cgk dgk]=series(A,B,C,D,Ak,Bk,Ck,Dk,);
[mc,pc]=bode(agk,bgk,cgk,dgk,1,w);
rlocus(agk,bgk,cgk,dgk)
```

系统的根轨迹图见图5-5。图5-5所示的系统根轨迹显示系统是条件稳定的，即当系统增益较低和较高时，系统都是不稳定的，用 rlocfind 命令可以求出系统稳定的增益范围是0.24～3.8。

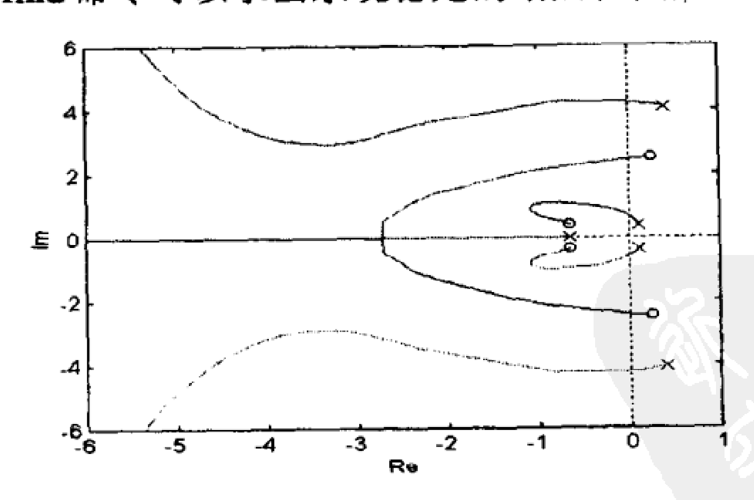


图 5-5　系统的根轨迹图

图5-6给出了系统的相频特性和幅频特性。

由伯德图可以得出系统的幅值裕量和相角裕量。采用下列

feedback 命令可求取闭环传递函数。

```
[At Bt Ct Dt]=feedkack(A, B, C, D, Ak, Bk, Ck, Dk);
[zT pT gainT]=ss2zp(At, Bt, Ct, Dt, 1)
```

从而得出系统的零极点是

```
zT=
   0.3975+4.0793j
   0.3975-4.0793j
  -14.3751
pT=
   -3.0000+3.0000j
   -3.0000-3.0000j
   -4
   -2
   -1.0000+1.0000j
   -1.0000-1.0000j
gainT=
      9.8000
```

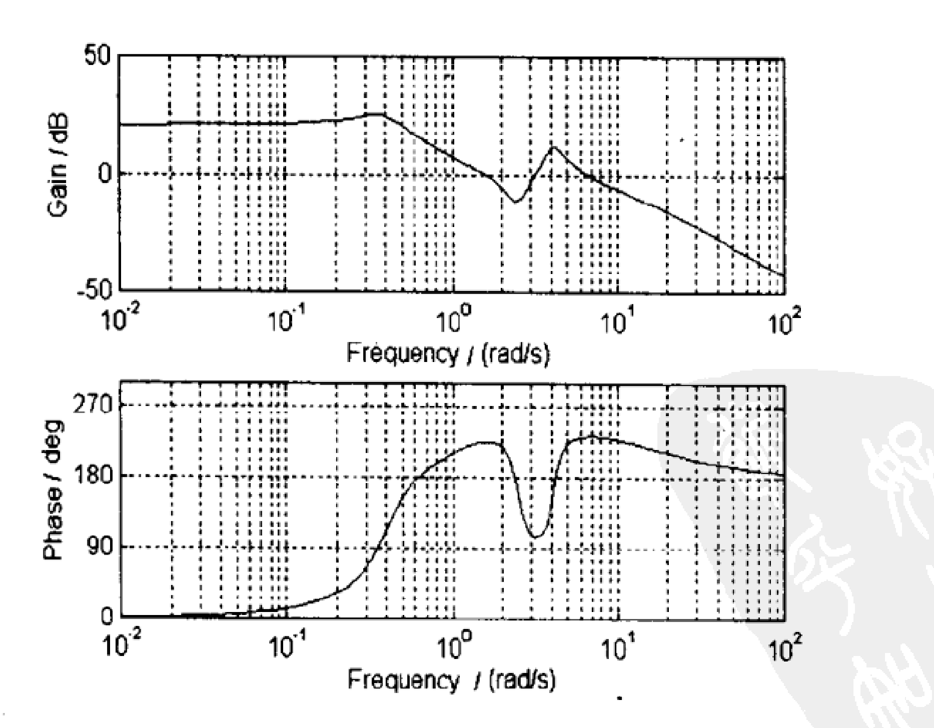


图 5-6 系统的伯德图

由上可见，闭环系统的零点是开环零点和补偿器极点之和，和其分离特性是吻合的。系统的阶跃响应可以由下列命令获得：

```
yc=step(At, Bt, Ct, Dt, 1, T);
dcg=dcgain(At, Bt, Ct, Dt);
yc=yc/dcg;
piot(t, yc)
```

图 5-7 给出了带观测器和状态反馈的系统的阶跃响应，从例 5-3 和例 5-2 可以看出，加入观测器降低了系统的稳定裕量，系统的过渡过程时间、达到峰值时间、超调量都略微有些上升。

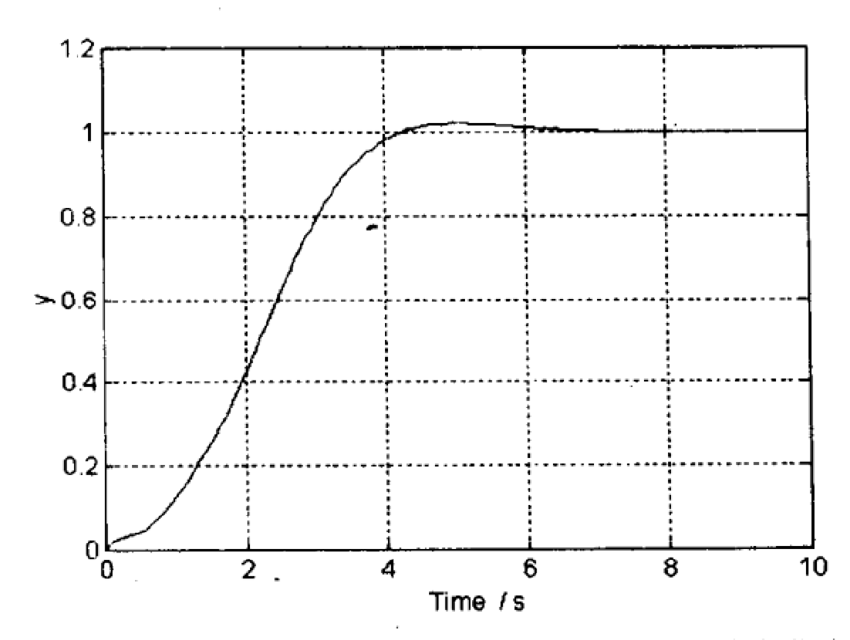


图 5-7　有着状态反馈和观测器的系统阶跃响应曲线

下面分析和验证观测器的估测是如何跟随系统实际状态的，在图 5-8 中，画出了实际状态和估测状态的动态响应曲线。**augstate** 命令用于将系统的实际状态和估测状态结合在一起，为具有典型性，选取系统的实际状态初始值与观测状态初始值是不相同的，分别选为[1　2　3]和[-1　-2　-3]，由图可见，实际状态和观测状态在 1.5s 后曲线基本吻合，即在不同的初始条件下在 1.5s 后，观测状态和实际状态达到一致。

系统设计及分析程序清单见本章附录中程序清单 2，注意给出相应的条件。

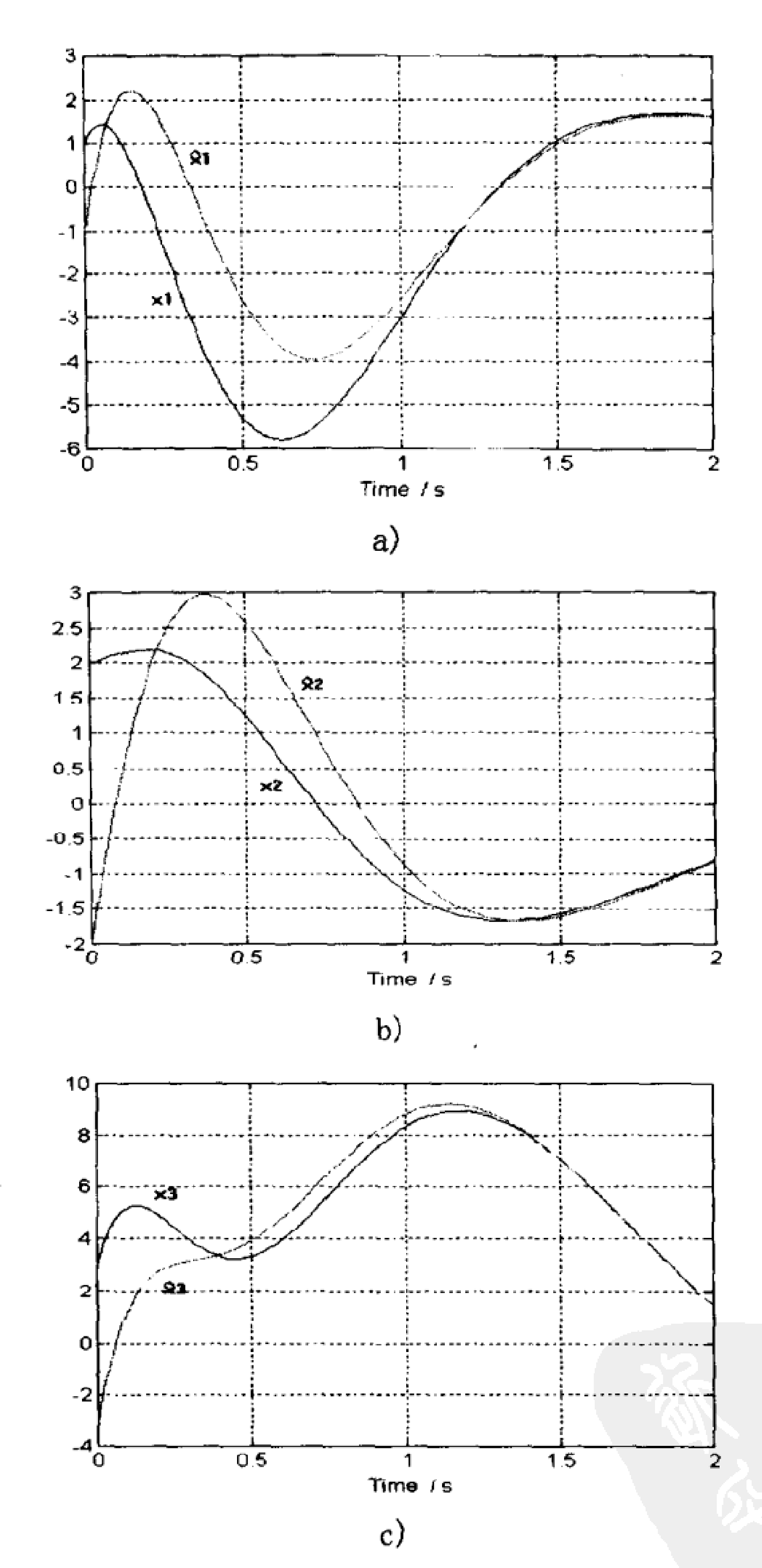


图 5-8　观测状态与实际状态的响应曲线

a）观测状态 $\hat{x}_1$ 与实际状态 x_1 的响应曲线　b）观测状态 $\hat{x}_2$ 与实际状态 x_2 的响应曲线

c）观测状态 $\hat{x}_3$ 与实际状态 x_3 的响应曲线

5.5 降阶观测器的设计

在前一节所介绍的观测器与系统的阶数相同，因此称之为全阶观测器。假如系统有 n 个状态，有 m 个可测量的输出，估测系统的的所有的状态是多余的，而且也是没必要的。理论上，需要估测的所有状态是未知的状态，是一个$(m-n)$维观测器，称之为降阶观测器。这种降阶观测器使补偿器更简化、更经济。 m 越大，效果越明显。

下面推导降阶观测器，首先定义状态的线性变换。

$$z = Tx$$

式中， T 的维数是$(n-m)\times n$； z 的维数是$(n-m)\times 1$。

T 保证 $E=\begin{bmatrix} C \\ T \end{bmatrix}$ 是非奇异的。

假如， C 的秩是 m ，定义 $E^{-1}=[P \vdots M]$，将 y 和 z 结合可得

$$\begin{bmatrix} y \\ z \end{bmatrix}=\begin{bmatrix} C \\ T \end{bmatrix}x \Rightarrow x=\begin{bmatrix} C \\ T \end{bmatrix}^{-1}\begin{bmatrix} y \\ z \end{bmatrix}=[P \vdots M]\begin{bmatrix} y \\ z \end{bmatrix}=Py+Mz \quad (5\text{-}28)$$

建立一个全阶观测器估测 z(观测器的阶数是$(n-m)$阶)，从而得到 x 的估测 $\hat{x}$ 。

$$\hat{x} = Py + M\hat{z} \quad (5\text{-}29)$$

式中， $\hat{z}$ 是 z 的估测值。

因此为了估测 x ，所需要做的是设计观测器估测 z ，因而应该知道 z 的动态性能，即 z 的一组差分方程。下面推导获得方程组的过程。

用 E 左乘原系统的状态方程，得

$$E\dot{x} = EAx + EBu$$

$$E\dot{x}=\begin{bmatrix} C \\ T \end{bmatrix}\dot{x}=\begin{bmatrix} C\dot{x} \\ T\dot{x} \end{bmatrix}=\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix}$$

$$EAx+EBu=\begin{bmatrix} C \\ T \end{bmatrix}Ax+\begin{bmatrix} C \\ T \end{bmatrix}Bu$$

$$=\begin{bmatrix} C \\ T \end{bmatrix}A[P \vdots M]\begin{bmatrix} y \\ z \end{bmatrix}+\begin{bmatrix} C \\ T \end{bmatrix}Bu$$

$$=\begin{bmatrix} CAP & CAM \\ TAP & TAM \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix}+\begin{bmatrix} CB \\ TB \end{bmatrix}u$$

由此得

$$\begin{bmatrix} \dot{y} \\ \dot{z} \end{bmatrix}=\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} y \\ z \end{bmatrix}+\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}u \tag{5-30}$$

式中

$$A_{11}=CAP \qquad A_{12}=CAM$$

$$A_{21}=TAP \qquad A_{22}=TAM$$

$$B_1=CB \qquad B_2=TB$$

因此 z 的差分方程是

$$\dot{z}=A_{22}z+\left(A_{21}y+B_2u\right) \tag{5-31}$$

参照上节全阶观测器的有关公式，加入误差校正项后 z 的观测器为

$$\dot{\hat{z}}=A_{22}\hat{z}+\left(A_{21}y+B_2u\right)+L\left(y-C\hat{x}\right) \tag{5-32}$$

式中，$(A_{21}y+B_2u)$ 项是已知的，可以认为是观测器的输入；$L(y-Cx)$是校正项，以使观测器误差系统稳定。由于$[P \mid M]$是 E 的逆，因此

$$\begin{bmatrix} C \\ T \end{bmatrix}[P \vdots M]=I$$

故有　$CP=I$， $CM=0$，所以

$$y-C\hat{x}=y-C\left(Py+M\hat{z}\right)=y-CPy-CM\hat{z}=0$$

校正项并没有起任何作用。在经典控制系统中，有时采用输出反馈是无效的，而采用输出反馈和输出变化率反馈才有效。理论上假如 y 是可测量的，则可假定 y 的变化率也是可测量的。因此，用可测量的输出的微分作为校正项。y 的差分方程是

$$\dot{y}=A_{11}y+A_{12}z+B_1u$$

将上述方程作为校正项，从而可得降阶观测器方程

$$\dot{\hat{z}}=A_{22}\hat{z}+(A_{12}y+B_2u)+L(\dot{y}-A_{11}y-B_1u-A_{12}\hat{z})$$

下面说明一下估测的误差是趋近于零的，也就是证明误差系统是稳定的。令

$$\tilde{z} = z - \hat{z}$$

根据上述方程可得误差系统方程为

$$\dot{\tilde{z}} = (A_{22} - LA_{12})\tilde{z} \tag{5-33}$$

在误差系统中含有一个未决定的矩阵 L，对于任意的观测器的特征值，是否对应存在矩阵 L。伦伯格证明：(C,A)矩阵的能观性等效于(A_{11},A_{12})的能观性。因此，由双重性，对于能观性的系统，在复平面上，误差系统的任意特征值，总可以对应地求得矩阵 L。应用 place 命令可以求得矩阵 L，命令格式为

```
L=place(A22', A12', op);    L=L'
```

在上述 y 的差分方程中，y 项可以通过简单的变量变换消除。定义 w 为

$$w = \hat{z} - Ly$$

将上述变换进行整理可得

$$\dot{w} = Fw + Dy + Gu \tag{5-34}$$

$$\hat{x} = Mw + Ny \tag{5-35}$$

其中

$$F = A_{22} - LA_{12} \qquad D = FL + A_{21} - LA_{11}$$

$$G = B_2 - LB_1 \qquad N = P + ML$$

当 C 有如下特殊形式时 T 的形式为

$$C = \left[I_m \ \vdots \ 0_{m\times(n-m)}\right]$$

$$T = \left[0_{(n-m)\times m} \ \vdots \ I_{n-m}\right] \qquad E = I_n$$

在这种情况下，不需要进行变换，A 和 B 可以直接地分为 A_{11}、A_{12}，可测量部分直接就是观测器的输出，因此，测量噪声被引入观测器。在这种情况下，设计降阶观测器必须注意噪声问题。

在降阶观测器中，分离特性仍然有效，这可以由下面闭环系统和误差系统的结合推导出

$$u = -Kx \qquad \dot{x} = Ax - BK\hat{x}$$

由方程式(5-28)和式(5-29)可得

$$\hat{x} = x - M\tilde{z}$$

故

$$\dot{x} = Ax - BKx + BKM\tilde{z} \tag{5-36}$$

由方程式(5-33)和式(5-36)得

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{z}} \end{bmatrix} = \begin{bmatrix} A - BK & BKM \\ 0 & A_{22} - LA_{12} \end{bmatrix} \begin{bmatrix} x \\ \tilde{z} \end{bmatrix}$$

由上式可见，闭环系统的特征值是控制器、观测器的特征值的结合。

由方程式(5-34)和式(5-35)得

$$u = -Kx = -KMw - KNy$$

$$\dot{w} = (F - GKM)w + (D - GKN)y$$

$$K(s) = \begin{bmatrix} F - GKM & D - GKN \\ -KM & -KN \end{bmatrix}$$

使用 MATLAB 语言中的 series 和 feedback 命令可以求取开环传递函数和闭环传递函数。

闭环传递函数可以将闭环系统和观测器方程结合起来，加入外部输入 v 求取，得

$$u = -K\hat{x} + \overline{N}v$$

$$\begin{bmatrix} \dot{x} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} A - BKNC & -BKM \\ DC - GKNC & F - GKM \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} B\overline{N} \\ G\overline{N} \end{bmatrix} v \tag{5-37}$$

$$T(s) = \begin{bmatrix} A - BKNC & -BKM & B\overline{N} \\ DC - GKNC & F - GKM & G\overline{N} \\ C & 0 & 0 \end{bmatrix} \tag{5-38}$$

【例 5-4】 降阶观测器设计。

在例 5-3 中，采用降阶观测器重新设计。假如状态变量 x_3 是可测量的，因此设计一个二阶观测器。观测器的极点为[-3+3j， -3-3j]。第一步选择 T，E 是非奇异的，因此是可逆的，P 和 M 可以由矩阵 E 中分离开，然后计算 A_{11}、A_{12} 等。下面是部分 MATLAB 程序：

```
op=[-3+3*j,-3-3*j];
```

```
T=[0 1 0;1 0 0];e=[C;T];
ei=inv(e);[mm nn]=size(C);
p=ei(:,1:mm);m=ei(:,mm+1:nn);
a11=C*A*p;a12=C*A*m;a21=T*A*p;
a22=T*A*m;b1=C*B;b2=T*B;
Lr=place(a22',a12',op);Lr=Lr';
```

下面为求取观测器参数和传递函数的程序

```
f=a22-Lr*a12;g=b2-Lr*b1;
d=f*Lr+(a21-Lr*a11);n=p+m*Lr;
Ak=f-g*k*m;Bk=d-g*k*n;Ck=k*m;Dk=k*n;
[zk pk gk]=ss2zp(Ak,Bk,Ck,Dk,1)
zk=
    -0.7141+0.4443j
    -0.7141-0.4443j
pk=
    2.8907+5.2744j
    2.8907-5.2744j
gk=
    1.56
```

由此可得补偿器的传递函数是

$$K_3(s)=\frac{1.56(s+0.71\pm j0.44)}{(s-2.89\pm j5.27)}$$

用下列程序求取系统的开环传递函数、根轨迹和伯德图：

```
[agk bgk cgk dgk]=series(A,B,C,D,Ak,Bk,Ck,Dk);
rlocus(agk, bgk, cgk, dgk);
bode(agk, bgk, cgk, dgk);
```

根轨迹图见图 5-9 。

伯德图见图 5-10 。

由伯德图和根轨迹图可以看出，系统是条件稳定的。

按下列程序求取系统的闭环传递函数、系统的零极点。

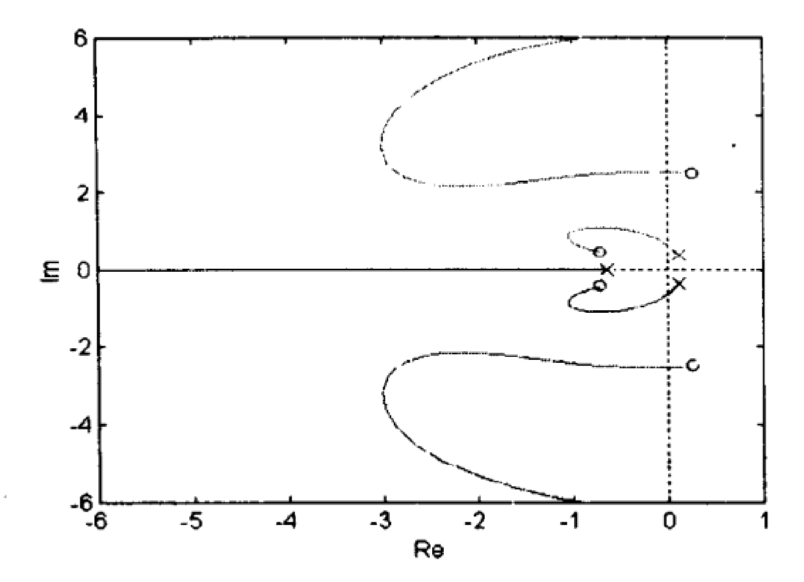


图 5-9 降阶观测器系统的根轨迹

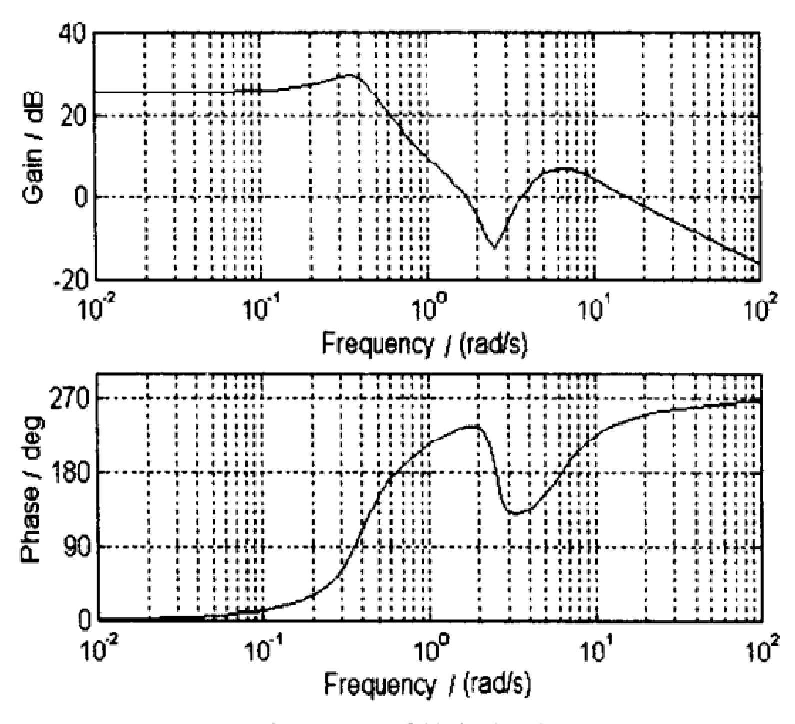


图 5-10 系统的伯德图

```
[At Bt Ct Dt]=feedback(A,B,C,D,Ak,Bk,Ck,Dk);
[zT pT gainT]=ss2zp(At,Bt,Ct,Dt,1);
zT =
     2.8907 + 5.2744i
     2.8907 - 5.2744i
     0.2500 + 2.4975i
```

```
    0.2500 - 2.4975i
pT =
    -3.0000 + 3.0000i
    -3.0000 - 3.0000i
    -2.0000
    -1.0000 + 1.0000i
    -1.0000 - 1.0000i
```

下列程序用来求取降阶观测器系统的阶跃响应曲线：

```
t=[0:0.1:9.9];
dcg=dcgain(At,Bt,Ct,Dt);
yc=step(At,Bt,Ct,Dt,1,t);
yc=yc/dcg;
plot(t,yc)
```

系统的阶跃响应曲线见图 5-11 。

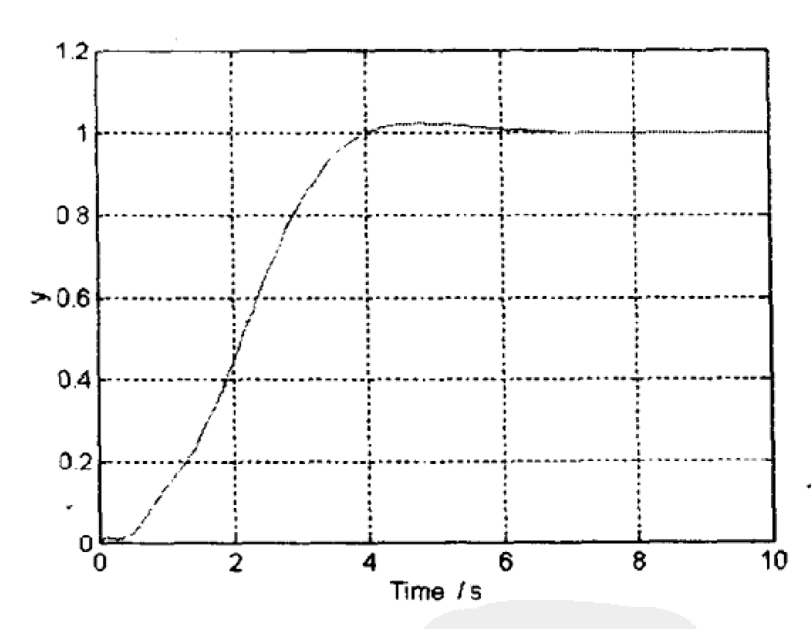


图 5-11　系统的阶跃响应曲线

由图可见，降阶观测器系统的阶跃响应和全阶观测器的阶跃响应是近似一致的。

在图 5-12 中画出了观测器状态的初始值和系统状态的初始值不一致时的阶跃响应；图中也给出了初始条件一致时的响应曲线。本章附录的程序 3 给出了上述观测器的设计步骤及程序清

单。

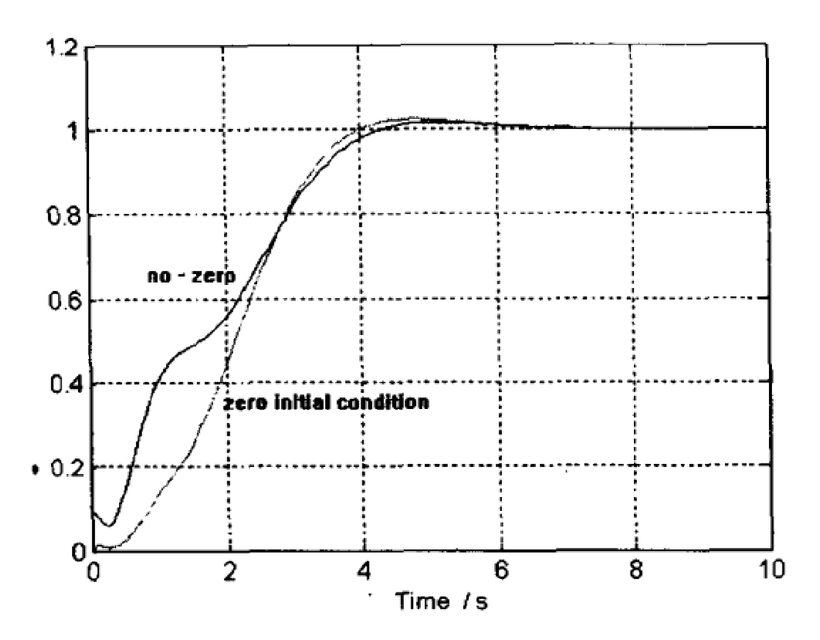


图 5-12　非零初始条件时的阶跃响应曲线

最后，分析一下系统的状态和估测状态的响应曲线。由方程式(5-35)得

$$\hat{x} = Mw + Ny = Mw + NCx$$

可将状态和估测状态分离后得

$$\begin{bmatrix} x \\ \hat{x} \end{bmatrix} = \begin{bmatrix} I & 0 \\ NC & M \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix}$$

系统的输出矩阵为

$$C_{t_} = [\text{eye}(3) \quad \text{zeros}(3,\ 2) \quad NC \quad M]$$

$$D_{t_} = \text{zeros}(6,\ 1)$$

选取系统状态变量的初始条件是 $x(0)=x_0=[1;\ 2;\ 3]$，估测状态的初始条件是 $xh_0=[-1;\ -2;\ -3]$。由于闭环系统的状态变量是 x 和 w，因此需要将 $\hat{x}$ 的初始条件 $\hat{x}_0$ 转换成 w 的初始条件 w_0。

$$w = \hat{z} - Ly = T\hat{x} - LCx$$

$$w_0 = T\hat{x}(0) - LCx(0)$$

图 5-13 给出了各状态变量和估测变量的阶跃响应曲线。第 3 个状态变量没有估测。

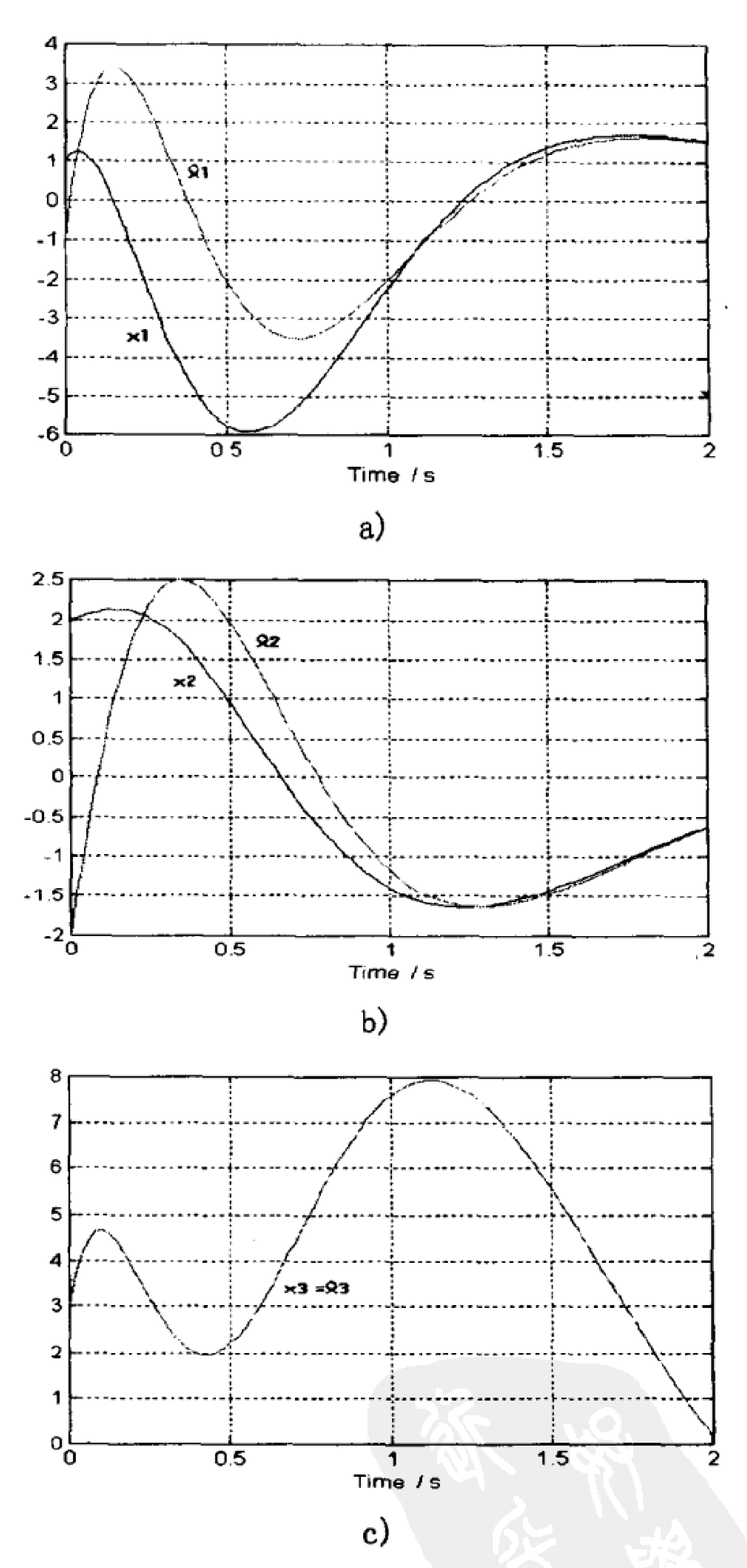


图 5-13 状态变量和估测变量的响应

a）状态变量 x_1 与估测变量 $\hat{x}_1$ 的响应曲线 b）状态变量 x_2 与估测变量 $\hat{x}_2$ 的响应曲线

c）状态变量 x_3 与估测变量 $\hat{x}_3$ 的响应曲线

由图 5-13 可以看出，状态变量和估测的状态变量的阶跃响应在 1.4s 左右达到一致。

5.6 有关状态空间设计的讨论

本章所讨论的方法，是一种有效的解决问题的方法，下面讨论一下极点配置的优点和限制。

主要的优点是极点的任意配置，无论开环零点和极点在什么位置，都可以任意配置期望的闭环极点。这为我们提供了控制时间响应的手段，假如系统的所有的状态都可以被测量和反馈的话，状态反馈可以提供简单而适当的设计。否则，可以采用观测器估测系统的状态。观测器增加了系统的复杂性，并且增加了整体费用，增加了更多的元件、连接，在数字系统中，更多的数字线降低了系统的可靠性，增加了维修费用。从控制系统的设计和性能上看，一般地说，观测器减少了系统的稳定裕量，它们也引入右半平面的极点或零点，导致了系统是条件稳定的，这是我们所不希望的。降阶观测器通常有较高的带宽，从而降低了对高频噪声的抑制。因此，选用观测器的类型是全阶还是降阶观测器必须非常慎重。通常，设计系统时必须综合地比较时间响应、稳态裕量、条件稳定性、带宽、补偿器阶数、元件数量、可靠性和敏感性等指标。

时域响应不仅取决于闭环极点，而且取决于闭环零点；然而前面的设计，并没有控制系统的零点。为达到零点配置，必须修正前面的方法，但却是很复杂的。另一个问题是补偿器的阶数与系统的阶数相同，这有时是不必要的。例如，一个高阶系统采用低阶的超前滞后补偿，也可以得到较好的效果。因此，应该首先用经典方法来补偿，如果结果不理想，再用状态空间方法设计。

在本章没有讨论的两个重要的问题是极点的选取和多输入多输出系统。极点的选取可以根据系统的性能指标来决定。而多输入多输出系统则会使反馈增益矩阵 $\boldsymbol{K}$ 变成矩阵而不是矢量，从而使答案是多种而不是唯一的。需要求取最优控制矩阵，这需

要应用某些 MATLAB 命令，这些命令在控制系统工具箱和鲁棒控制工具箱中。

特别要注意的是，由于状态空间方法是时域设计方法，一般不能满足频域指标。事实上，补偿器观测器系统是易于不稳定的，对于模型的变化是非常敏感的，不具有鲁棒性。可以采用鲁棒控制等先进的方法解决系统的鲁棒性问题。

习　题

5-1 考虑如下系统：

$$A=\begin{bmatrix}0&1&0\\0&0&1\\2&0&-1\end{bmatrix}\quad B=\begin{bmatrix}1\\2\\0\end{bmatrix}\quad C=\begin{bmatrix}1&0&0\end{bmatrix}\quad D=0$$

(1) 系统的期望闭环极点在[-10, -1+j, -1-j]，求系统的状态反馈增益矩阵。

(2) 求补偿器的等效传递函数、根轨迹、伯德图和闭环系统的阶跃响应。

5-2 对上题所述系统，求

(1) 设计全阶观测器，选取观测器的极点为[-40, -4+4j, -4-4j]，求等效闭环传递函数、根轨迹、伯德图和系统的阶跃响应。

(2) 设计降阶观测器，其它同(1)。

(3) 假设观测器的极点在[-40, -1, -2]，重复(1)。

(4) 假设观测器的极点在[-1, -2]，重复(2)。

附录　程序清单

1. 设计状态反馈的程序　其 MATLAB 程序清单如下：

```
% Observer  design
A=[-0.4 0 -0.01;1 0 0;-1.4 9.8 -0.02];
B=[6.3;0;9.8];
C=[0 0 1];D=0;
w=[0.01 10];t=0:0.05:10;
P=[-2 -1+j -1-j];
K=place(A,B,P);
```

```
bode(A,B,K,0);
[gm pm wpc wgc]=margin(A,B,K,0)
figure;rlocus(A,B,K,0);
At=A-B*K;Bt=B;Ct=K;Dt=D;
[zT pT gainT]=ss2zp(At,Bt,Ct,Dt);
dcg=dcgain(At,Bt,Ct,Dt);
figure
yc=step(At,Bt,Ct,Dt,1,t);yc=yc/dcg;
plot(t,yc);
grid
end
```

2. 设计全阶观测器的程序　其 MATLAB 程序清单如下：

```
% Observer  design
A=[-0.4 0 -0.01;1 0 0;-1.4 9.8 -0.02];
B=[6.3;0;9.8];
C=[0 0 1];D=0;
w=[0.01 10];t=0:0.05:10;
P=[-2 -1+j -1-j];
op=[-4 -3+3*j -3-3*j];
x0=[1 2 3];xh0=[-1 -2 -3];
K=place(A,B,P);
L=place(A',C',op);L=L';
Ak=A-B*K-L*C;Bk=L;Ck=K;
[tmp1 tmp2]=size(D);
Dk=zeros(tmp1,tmp2);
[zk pk gaink]=ss2zp(Ak,Bk,Ck,Dk,1);
[Ac Bc Cc Dc]=series(A,B,C,D,Ak,Bk,Ck,Dk);
rlocus(Ac,Bc,Cc,Dc);
axis([-6 1 -6 6]);
figure
```

```
[mc pc]=bode(Ac,Bc,Cc,Dc,1,w);
[gm pm wpc wgc]=margin(mc,pc,w);
[At Bt Ct Dt]=feedback(A,B,C,D,Ak,Bk,Ck,Dk);
[zt pt gaint]=ss2zp(At,Bt,Ct,Dt,1);
dcg=dcgain(At,Bt,Ct,Dt);
yc=step(At,Bt,Ct,Dt,1,t);yc=yc/dcg;
figure
plot(t,yc);
grid
[ata bta cta dta]=augstate(At,Bt,Ct,Dt);
t2=[0:0.02:2]';
x_xh=initial(ata,bta,cta,dta,[x0 xh0],t2);
x=x_xh(:,2:4);xh=x_xh(:,5:7);
figure;plot(t2,[x(:,1) xh(:,1)]);grid
figure;plot(t2,[x(:,2) xh(:,2)]);grid
figure;plot(t2,[x(:,3) xh(:,3)]);grid
end
```

3. 设计降阶观测器的程序 其 MATLAB 程序清单如下：

```
% Observer  design
A=[-0.4 0 -0.01;1 0 0;-1.4 9.8 -0.02];
B=[6.3;0;9.8];
C=[0 0 1];D=0;
w=[0.01 10];t=0:0.05:10;
p=[-2 -1+j -1-j];
op=[-3+3*j -3-3*j];
x0=[1 2 3];xh0=[-1 -2 -3];
k=place(A,B,p);
T=[0 1 0;1 0 0];
e=[C;T];
ei=inv(e);[mm nn]=size(C);
```

```
p=ei(:,1:mm);m=ei(:,mm+1:nn);
a11=C*A*p;a12=C*A*m;a21=T*A*p;
a22=T*A*m;b1=C*B;b2=T*B;
Lr=place(a22',a12',op);Lr=Lr';
f=a22-Lr*a12;g=b2-Lr*b1;
d=f*Lr+(a21-Lr*a11)
n=p+m*Lr;
Ak=f-g*k*m;Bk=d-g*k*n;Ck=k*m;Dk=k*n;
[zk pk gk]=ss2zp(Ak,Bk,Ck,Dk,1);
zk,pk,gainT
[Ac Bc Cc Dc]=series(A,B,C,D,Ak,Bk,Ck,Dk);
rlocus(Ac,Bc,Cc,Dc);
axis([-6 1 -6 6]);
figure
[mc pc]=bode(Ac,Bc,Cc,Dc,1,w);
bode(Ac,Bc,Cc,Dc);
[gm pm wpc wgc]=margin(mc,pc,w)
[At Bt Ct Dt]=feedback(A,B,C,D,Ak,Bk,Ck,Dk);
[zt pt gaint]=ss2zp(At,Bt,Ct,Dt,1);
dcg=dcgain(At,Bt,Ct,Dt);
yc=step(At,Bt,Ct,Dt,1,t);yc=yc/dcg;
figure
plot(t,yc,'g')
hold;
At1=A-B*K;Bt1=B;Ct1=C;Dt1=D;
dcg1=dcgain(At1,Bt1,Ct1,Dt1);
yc1=step(At1,Bt1,Ct1,Dt1,1,t);yc1=yc1/dcg1;
plot(t,yc1,'r')
grid
ycc=lsim(At,Bt,Ct,Dt,ones(201,1),t,[x0 xh0(1:2)]);
```

```
ycc=ycc/dcg;
[zt pt gt]=ss2zp(At,Bt,Ct,Dt,1);
zt,pt
Ct_=[eye(nn) zeros(nn,nn-mm);n*C m];
Dt_=zeros(2*nn,1);
t2=[0:0.02:2]';
w0=T*xh0'-Lr*C*x0';w0=w0';
x_xh=initial(At,Bt,Ct_,Dt_,[x0 w0],t2);
figure;plot(t2,[x_xh(:,1) x_xh(:,4)]);grid
figure;plot(t2,[x_xh(:,2) x_xh(:,2)]);grid
figure;plot(t2,[x_xh(:,3) x_xh(:,3)]);grid
end
```

第6章　数字控制系统设计方法

6.1　概述

在控制系统中，根据信号的传递和变换方式，控制系统分为连续控制系统和离散控制系统两大类。若系统中各处的信号是时间的连续函数，则称该系统为连续控制系统；若系统中一处或几处的信号不是时间的连续函数，而是一系列的脉冲或数码，则称之为离散控制系统，或称为数字控制系统。

一种典型的数字控制系统见图 6-1

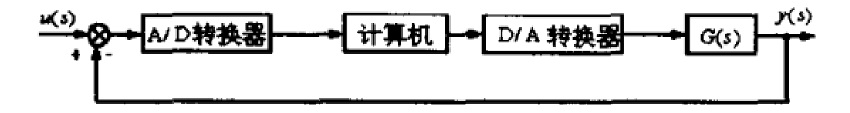


图 6-1　数字控制系统

通常，连续的误差信号由模数转换器转换成数字量，计算机实时地采集并处理该信号，通过大量的运算，提供控制信号。由于大部分的设备是连续的，计算机的输出控制信号由数模转换器转换成连续变化的模拟信号。

同在连续控制系统中一样，控制工程师必须为数字控制系统设计补偿器，补偿器被编制成一组差分方程输入到计算机里。数字系统的设计技术与连续系统的设计技术是同步发展的，对单输入单输出系统可以用根轨迹或伯德图的转换技术进行设计。而数字系统设计比连续系统设计更复杂。

首先，我们讨论一些基本的数字控制系统分析方法。

6.2　差分方程

连续系统的动态性能是由微分方程组描述的，而数字控制系统对系统中变量的测量是不连续的，只能测得这些变量在采样时

刻 0 、 T 、 $2T$ 、 $3T$ 、…、 nT 的数值，因此，数字控制系统的动态性能是用差分方程描述的。

例如，二次差分方程如下：

$$y[(k+2)T]-0.7y[(k+1)T]+0.1y[kT]=10x[(k+2)T]$$

差分方程的编程是非常容易的，可以将上式改写为

$$y[(k+2)T]=0.7y[(k+1)T]-0.1y[kT]+10x[(k+2)T]$$

为解这个方程，我们仅需要两个寄存器，用 $y(0)$、 $y(1)$给寄存器赋值，然后写出如下程序：

```
STORE  Y(0)  IN  RIGISTER  A
STORE  Y(1)  IN  RIGISTER  B
1. Y=10X-0.1A+0.7B
2. STORE  B  IN  A
3. STORE  Y  IN  B
4. RETURN  TO  1
```

从而可以得到 y 的终值。

差分方程和微分方程的关系由下面的推导可以得出，系统的微分方程如下：

$$\frac{\mathrm{d}y}{\mathrm{d}t}+5y=x(t)$$

微分项在连续系统中是速度。然而，假如仅仅观察系统在每个 T 秒时刻的位置，我们如何测量系统的速度呢？最简单的方法是假定在测量期间，速度是不变的常数，则

$$\frac{\mathrm{d}y}{\mathrm{d}t}=\frac{y[(k+1)T]-y[kT]}{T}$$

假定当前是在$(k+1)T$ 时测得的，则微分方程可以近似为

$$\frac{y[(k+1)T]-y[kT]}{T}+5y[(k+1)T]=x[(k+1)T]$$

整理后得

$$y[(k+1)T]-\frac{1}{1+5T}y[kT]=\frac{T}{1+5T}x[(k+1)T]$$

这是典型的差分方程形式。由已知的输入函数和输出函数的

初始条件，就可求得差分方程的解。

很明显，所使用的速度表达式的近似形式决定了差分方程的形式，在本章的后面，将讨论几种不同的速度近似形式，以及它们的优缺点。

由上述差分方程可以看到，一系列的输出值是由前几次的输出与现在及过去的输入值组成的。这种方法存在着一个问题，即不能得到解的闭合形式，因此难以预测系统的性能和稳定性。

解差分方程可以用在时域内解微分方程的方法，差分方程的解也是由自由分量和强制分量组成的，可以使用频率转换技术。最后，系统可以被写成状态空间的形式，得到脉冲响应的解。

下面举一个解差分方程的例子，求解其自由分量，由此观察数字系统的稳定性。

一阶齐次差分方程为

$$y_{k+1} - ay_k = 0 \tag{6-1}$$

初始条件为

$$y_0 = 1$$

式中，y_{k+1} 是 $y[(k+1)T]$ 的简写形式，用 r_n 代替 y_{k+n} 可以形成特征方程，上式的特征方程是

$$r-a=0 \qquad r=a$$

特征方程实数根的自由分量是 r^k，差分方程的解是

$$y_k = a^k \qquad (k=0, 1, 2, \cdots) \tag{6-2}$$

由上式可见，假如 a 的幅值小于 1，自由分量趋向于 0，系统是稳定的；若 a 的幅值大于 1，系统是不稳定的。当 a 的值为复数时，则其虚部决定了系统的动态响应性能，尽管我们考虑的是一阶系统，当 a 为复数时，系统的响应是振荡的。而一阶连续系统没有振荡响应的现象，这是数字系统和连续系统的差别。

一个典型的二阶齐次差分方程是

$$y_{k+2} - ay_{k+1} + by_k = 0 \tag{6-3}$$

由于我们只对解的自由分量感兴趣，因此不考虑方程的初始条件，上述系统的特征方程是

$$r^2 - ar + b = 0 \tag{6-4}$$

二阶特征方程的解可以有如下几种情况：

(1)两个完全不同的实数解，这种情况其响应是过阻尼的，$r = r_1$、r_2，则

$$y_k = c_1 r_1^k + c_2 r_2^k \tag{6-5}$$

(2)两个完全相同的实数，其响应是临界阻尼的，$r = r_1$、r_2，则

$$y_k = (c_1 + c_2 k) r_1^k \tag{6-6}$$

(3)解是复数，其响应是欠阻尼的。

$$r = \alpha \pm \mathrm{j}\beta = R\mathrm{e}^{\mathrm{j}\pm\theta} \tag{6-7}$$

则

$$y_k = R^k (c_1 \cos\theta k + c_2 \sin\theta k) \tag{6-8}$$

由上式可以清楚地看到，只有当 r 的幅值小于 1 时，系统才稳定。

6.3 采样信号的频谱

对连续信号 $f(t)$ 每隔 T 秒采样一次，将连续时间函数 $f(t)$ 变换成离散时间函数 $f^*(t)$，其采样过程见图 6-2 。

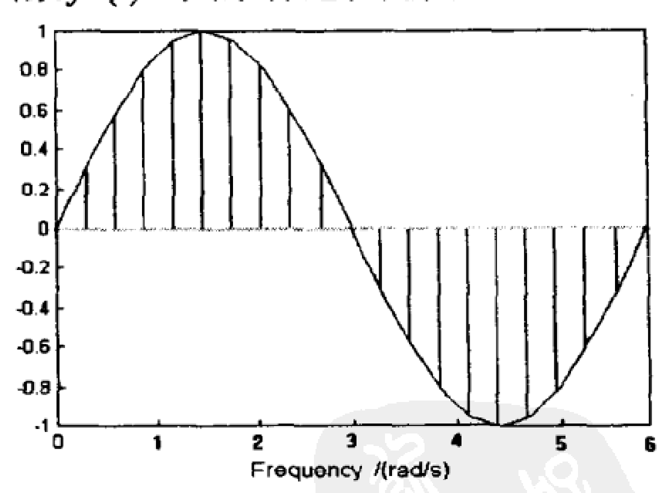


图 6-2　信号和采样

在每一个 T 秒，开关闭合无限短的时间，T 为采样周期，则离散时间函数可以表示为

$$f^*(t) = f(kT) \qquad (k = \cdots,-2,-1,0,+1,+2,\cdots) \tag{6-9}$$

也可以把离散信号用数学表达式表示成脉冲列与原来连续信号的乘积，这种方式称之为脉冲调制，即

$$f^*(t)=f(t)\sum_{k=-\infty}^{+\infty}\delta(t-kT) \tag{6-10}$$

因为脉冲列是周期函数，可以将其表示为傅里叶级数

$$\sum_{k=-\infty}^{+\infty}\delta(t-kT)=\frac{1}{T}\sum_{n=-\infty}^{+\infty}\mathrm{e}^{\mathrm{j}(2\pi n/T)t}$$

采样信号的傅里叶变换为

$$F^*(\omega)=\int_{-\infty}^{+\infty}[f(t)\frac{1}{T}\sum_{n=-\infty}^{+\infty}\mathrm{e}^{\mathrm{j}(2\pi n/T)t}]\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$$

经整理得

$$F^*(\omega)=\frac{1}{T}\sum_{n=-\infty}^{+\infty}\int_{-\infty}^{+\infty}f(t)\ \mathrm{e}^{-\mathrm{j}(\omega-2\pi n/T)t}\ \mathrm{d}t$$

由于原始连续信号的傅里叶变换为

$$F(\omega)=\int_{-\infty}^{+\infty}f(t)\ \mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$$

则采样信号的傅里叶变换为

$$F^*(\omega)=\frac{1}{T}\sum_{n=-\infty}^{+\infty}F(\omega-2\pi n/T) \tag{6-11}$$

图 6-3 表示了连续信号和离散信号频率项之间的关系。

采样信号的频谱是连续信号频谱的周期响应。这可以推导出采样系统的一些问题。

6.3.1 采样定理

连续信号 $f(t)$经采样后变成 $f^*(t)$，那么 $f^*(t)$能否恢复成与 $f(t)$一样的连续信号呢？事实上，由 $f(t)$到 $f^*(t)$的采样过程总是近似的，采样频率越高，两者越近似，当采样频率趋近于无穷大时，$f^*(t)$与 $f(t)$就完全一样了。但是，提高采样频率 f，也就是减小采

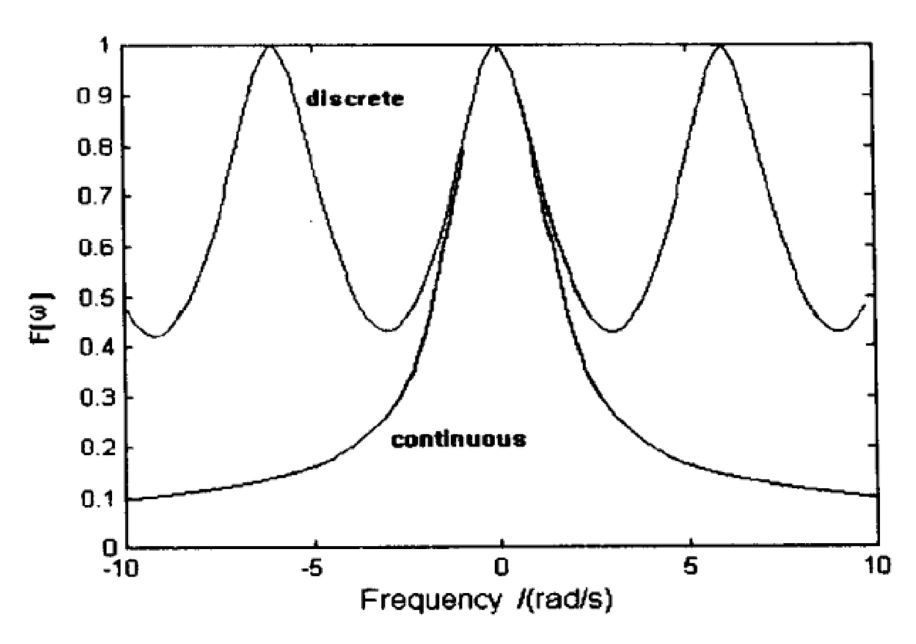


图 6-3　连续信号和离散信号的频率响应

样周期 T。采样周期是计算机对有关信息进行加工处理的时间，T越小，要求计算机的工作的速度越高，这是有一定限度的。当采样频率过低时，$f(t)$与 $f^*(t)$就会不近似，即丢失应有的信息，因此采样频率既不能太高，也不能太低，否则不能用低通滤波器重构原来的信号 $f(t)$。

香农采样定理给出了如下的采样频率的限制：

$$\frac{2\pi}{T} \geqslant 2\omega_{max}$$

或

$$\omega_s \geqslant 2\omega_{max}$$

式中，ω_s是采样角频率；ω_{max}是与原始信号相关的最高频率。

如果对一个具有有限频谱的连续信号 $f(t)$进行采样，当$\omega_s \geqslant 2\omega_{max}$时，离散信号可以无失真地再现原连续信号 $f(t)$。

当$\omega_s < 2\omega_{max}$时，即使采用理想滤波器，也无法在滤波器的输出端得到原连续信号 $f(t)$。我们称采样频率$\omega_s = 2\omega_{max}$为奈奎斯特频率。

6.3.2 信号的混叠

设连续信号 $f(t)$的傅里叶变换为 $F(j\omega)$，具有图 6-4a 所示的

频谱，离散信号 $f^*(t)$的傅里叶变换为

$$F^*(j\omega)=\frac{1}{T}\sum_{n=-\infty}^{+\infty}F(j\omega+jn\omega_s)$$

其展开式为

$$F^*(j\omega)=\frac{1}{T}\{F(j\omega)+F[j(\omega+\omega_s)]+F[j(\omega-\omega_s)]+$$

$$F[j(\omega+2\omega_s)]+F[j(\omega-2\omega_s)]+\cdots\}$$

由其展开式可以看到，$F^*(j\omega)$是以$2\pi/\omega_s$为周期的周期函数。由此可以绘制出离散信号的频谱$|F^*(j\omega)|$。当$\omega_s<2\omega_{max}$时，$f(t)$采样后的信号频谱$|F^*(j\omega)|$各高频分量和低频分量之间产生混叠现象，见图 6-4b 。采样信号的频谱在ω处的值，是模拟信号的频谱在频率$\omega+n\omega_s$处所有值之和，因此经采样，不可能区分出这些频率点上的值的大小。当$\omega_s\geqslant 2\omega_{max}$时，$f^*(t)$的频谱见图 6-4c ，离散信号的频谱是由无穷多个形状与原连续信号频谱相同的孤立频谱构成的离散频谱，其中 n=0 对应的频谱与连续信号的频谱是近似的，只是在幅值上差 T 倍，而其它各个孤立频谱是由于采样而产生的高频频谱。由上述分析可以得出，提高采用频率使之高于奈奎斯特频率，可以减少混叠。

混叠产生另一个不能靠提高采样频率而解决的问题。系统中的噪声是高频的，在连续系统中，假如噪声频率在控制频带以外，它对系统的响应没有影响。然而，在采样系统中，当采样噪声信号时，噪声频谱会分布在频率轴的上下部分，即假如噪声在频率ω_D中心，采样信号将含有$\omega_D-2\pi n/T$的频率，无论选取什么样的采样频率，在控制频带内都会得到噪声信号。

为了避免边带重叠和高频噪声问题，我们常常用一个低通滤波器置于采样器之前。这个滤波器叫反混叠滤波器。它降低了高频信号部分，使得前面讨论的问题最小化了。假如控制系统设计使用反混叠滤波器，在仿真系统中也一定要包含它，以保证仿真的正确性。

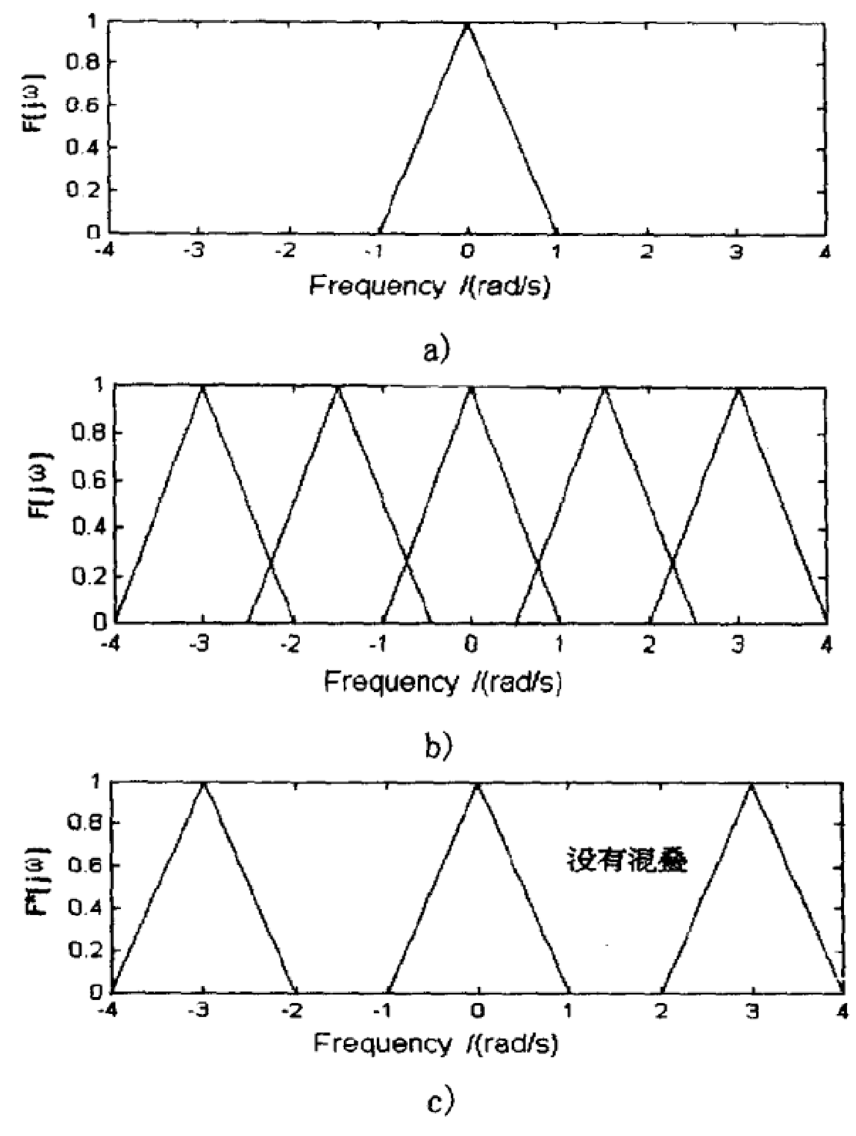


图 6-4　信号的频谱

a) 连续信号频谱　b) $\omega_s < 2\omega_{max}$时的频谱　c) $\omega_s \geq 2\omega_{max}$时的频谱

6.4　z 变换

z 变换是分析设计离散系统的重要工具之一，它在离散系统中的作用与拉普拉斯变换在连续系统中的作用是相似的。

设连续信号$f(t)$的拉普拉斯变换为$F(s)$，$f(t)$经采样开关后变为离散时间函数$f^*(t)$，采样周期为T，设$T<0$时，$f(t)=0$，则

$$f^*(t)=\sum_{k=0}^{+\infty} f(kT)\delta(t-kT) \tag{6-12}$$

对式(6-12)进行拉普拉斯变换得

$$F^*(s)=\sum_{k=0}^{+\infty}f(kT)\mathrm{e}^{-skT}$$

采样信号的拉普拉斯变换是 s 的超越函数，为了方便，定义一个新的变量 z，令

$$z=\mathrm{e}^{sT}$$

由此将 $F^*(s)$的函数变为 z 的函数，则采样信号的 z 变换定义为

$$F(z)=\sum_{k=0}^{\infty}f(kT)z^{-k} \tag{6-13}$$

若式(6-13)的级数收敛，则称 $F(z)$为 $f^*(t)$的 z 变换。记作 $F(z)=Z[f^*(t)]$。

z 变换实际上是离散时间函数的拉普拉斯变换，只能反映连续时间函数在采样时刻的值，因此，如果两个信号在采样点上具有相同的值，则其 z 变换是相同的。

一般信号的 z 变换可以查阅信号处理手册，z 变换的最重要的意义是将微分方程转化为代数方程，如

$$\begin{aligned}
&f[(k+1)T] \;\rightarrow\; zF(z)-zf(0)\\
&f[(k+2)T] \;\rightarrow\; z^2F(z)-z^2f(0)-zf(1)\\
&\qquad\vdots \qquad\qquad\qquad \vdots\\
&f[(k+n)T] \;\rightarrow\; z^nF(z)-z^nf(0)-z^{n-1}f(1)-\cdots-zf(n-1)
\end{aligned}$$

例如，我们可以将下式零状态的差分方程变换如下：

$$y_{k+3}+0.3y_{k+2}-y_{k+1}-0.05y_k=5x_{k+1}+x_k$$

$$(z^3+0.3z^2-z-0.05)Y(z)=(5z+1)X(z)$$

推出 $Y(z)$，得

$$Y(z)=\frac{5z+1}{z^3+0.3z^2-z-0.05}X(z) \tag{6-14}$$

给出输入的 z 变换，我们可用 z 反变换的方法得到 y_k。可以采用部分分式展开的方法，首先由 $Y(z)$除以 z，然后展开，在展开式的各项系数已经确定后，再乘以 z，这种处理方法的目的是

为了得到 $z/(z-a)$形式的项，查表可以得到输出解。

当然，控制工程师更感兴趣的是系统的性能，而不是输出形式，系统性能是由传递函数决定的，式(6-14)的传递函数为

$$H(z)=\frac{Y(z)}{X(z)}=\frac{5z+1}{z^3+0.3z^2-z-0.05}$$

由 z 域传递函数的框图组成的控制系统，与由串并联反馈组成的连续系统都服从相同的规则。因此梅森规则同样适用于采样系统。

6.5 离散状态空间模型

离散状态空间模型是离散系统时域描述的一种表达形式，其一般形式为

$$\boldsymbol{x}_{k+1}=\boldsymbol{A}\boldsymbol{x}_k+\boldsymbol{B}\boldsymbol{u}_k \tag{6-15}$$

$$\boldsymbol{y}_k=\boldsymbol{C}\boldsymbol{x}_k+\boldsymbol{D}\boldsymbol{u}_k \tag{6-16}$$

状态方程可以导出 z 变换的模型。在零初始条件下，z 变换的状态空间方程为

$$z\boldsymbol{X}(z)=\boldsymbol{A}\boldsymbol{X}(z)+\boldsymbol{B}\boldsymbol{U}(z) \tag{6-17}$$

$$\boldsymbol{Y}(z)=\boldsymbol{C}\boldsymbol{X}(z)+\boldsymbol{D}\boldsymbol{U}(z) \tag{6-18}$$

由方程式(6-17)解 $X(z)$得

$$\boldsymbol{X}(z)=(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}\boldsymbol{U}(z)$$

把 $X(z)$代入到输出方程式(6-18)中，有

$$\boldsymbol{Y}(z)=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}]\boldsymbol{U}(z) \tag{6-19}$$

$$\boldsymbol{Y}(z)=\boldsymbol{G}(z)\boldsymbol{U}(z) \tag{6-20}$$

因此，传递函数为

$$\boldsymbol{G}(z)=[\boldsymbol{C}(z\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}]=\boldsymbol{C}\boldsymbol{\Phi}(z)\boldsymbol{B}+\boldsymbol{D} \tag{6-21}$$

式中 $\boldsymbol{\Phi}(z)=(z\boldsymbol{I}-\boldsymbol{A})^{-1}$

状态方程、差分方程和脉冲传递函数之间是等价的，因此是可以相互转换的。MATLAB 中许多命令都是建立在状态空间模型下，因此常常把状态空间模型和传递函数模型相互转换，其对

应的转换在 MATLAB 中有标准化子模块来完成，大大节省了用户的时间，方便了用户的使用。转换命令有 tf2ss 、 ss2zp 和 ss2tf，其中

tf2ss：将传递函数形式转换成状态空间形式。

ss2zp：将状态空间形式转换成 z 变换形式。

ss2tf：将状态空间形式转换为传递函数形式。

其转换方法及举例将在后面介绍。

6.6 数字控制系统仿真

数字控制系统通常既包括数字单元，又包括连续单元。在频域中，数字单元有 z 变换模型，连续单元有拉普拉斯变换模型，因此对于一个系统，必须使两者协调一致。在数字控制系统中的频域设计技术必须被时域的系统仿真所验证。 SIMULINK 允许用户将数字单元和连续单元模型化。

当将数字系统模型化时，就把由 $G(s)$表示的连续系统传递函数变换成“等效的”数字系统传递函数 $G(z)$。在这里提到“等效的”是因为对于 $G(z)$而言，不必与对应的 $G(s)$精确匹配。假如数字模型与连续模型产生相同的脉冲响应，它们的阶跃响应就不一致；假如阶跃响应一致的话，则脉冲响应就不一致。数字系统模型在频域内的响应与连续系统模型在频域内的响应不是精确对应的。

将连续系统离散化有一些方法，其中有时域响应匹配法、数字差分法(如后向差分法、前向差分法)、双线性变换法。每种方法都有它的优点和不足，正如前面提到的，一些方法保护了时域响应，而另外一些方法则更好地保护了频域响应。从控制的观点出发，保护稳定性是非常重要的，一些技术方法保护了稳定性，而另外一些却没有，那些不保护系统稳定性的方法必须慎用，否则将会产生错误的仿真结果。

6.6.1 脉冲响应不变法

脉冲响应不变法，用对照连续系统和离散系统的脉冲响应，

将连续系统变换成离散系统。考虑如下的连续系统传递函数和它的脉冲响应：

$$H(s)=\frac{10}{s+5} \qquad h(t)=10\mathrm{e}^{-5t}u(t)$$

脉冲响应的采样形式和它的 z 变换是

$$h(nT)=10\mathrm{e}^{-5nT}u(nT)$$

$$H(z)=\frac{10z}{z-\mathrm{e}^{-5T}}$$

当用数学模型表示一个连续系统时，用于保护脉冲响应的步骤如下：

从 $H(s)$得到脉冲响应；令 $t=nT$ ，从而将连续响应变换成离散响应；将离散脉冲响应变换成 $H(z)$。

这种变换也可应用于状态空间形式，连续系统的脉冲响应为

$$h(t)=\boldsymbol{C}\mathrm{e}^{\boldsymbol{A}t}\boldsymbol{B}+\boldsymbol{D}\delta(t)$$

离散系统的脉冲响应为

$$h(nT)=\boldsymbol{C}_{\mathrm{d}}\boldsymbol{A}_{\mathrm{d}}^{n-1}\boldsymbol{B}_{\mathrm{d}}+(\boldsymbol{D}_{\mathrm{d}}-\boldsymbol{C}_{\mathrm{d}}\boldsymbol{A}_{\mathrm{d}}^{-1}\boldsymbol{B}_{\mathrm{d}})\delta(nT)$$

令 $t=nT$，对 $h(t)$采样，以便将连续系统响应变换成离散响应，比较两种响应，可以推导出

$$\boldsymbol{A}_{\mathrm{d}}=\mathrm{e}^{\boldsymbol{A}T} \qquad \boldsymbol{B}_{\mathrm{d}}=\mathrm{e}^{\boldsymbol{A}T}\boldsymbol{B} \qquad \boldsymbol{C}_{\mathrm{d}}=\boldsymbol{C} \qquad \boldsymbol{D}_{\mathrm{d}}=\boldsymbol{D}+\boldsymbol{C}\boldsymbol{B}$$

可以写出一个函数，以便完成这种变换。

6.6.2 带有零阶保持器的 z 变换法

实际上，离散信号从不直接地去驱动一个连续系统。离散信号首先经过 D/A 转换器， D/A 转换器产生与离散信号对应的连续输出，最简单和常用的 D/A 转换器是把计算机输出的二进制信号转换成电平电压，并保持此值直到计算机输出下一个值。这个装置被称作零阶保持器(ZOH)，零阶保持器输出的波形见图 6-5 。

零阶保持器将每一个输入单位脉冲信号变换成脉冲输出，由于每一步是由单位脉冲积分而来，可以得到零阶保持器的传递函

数(单位脉冲在周期 T 的拉普拉斯变换)：

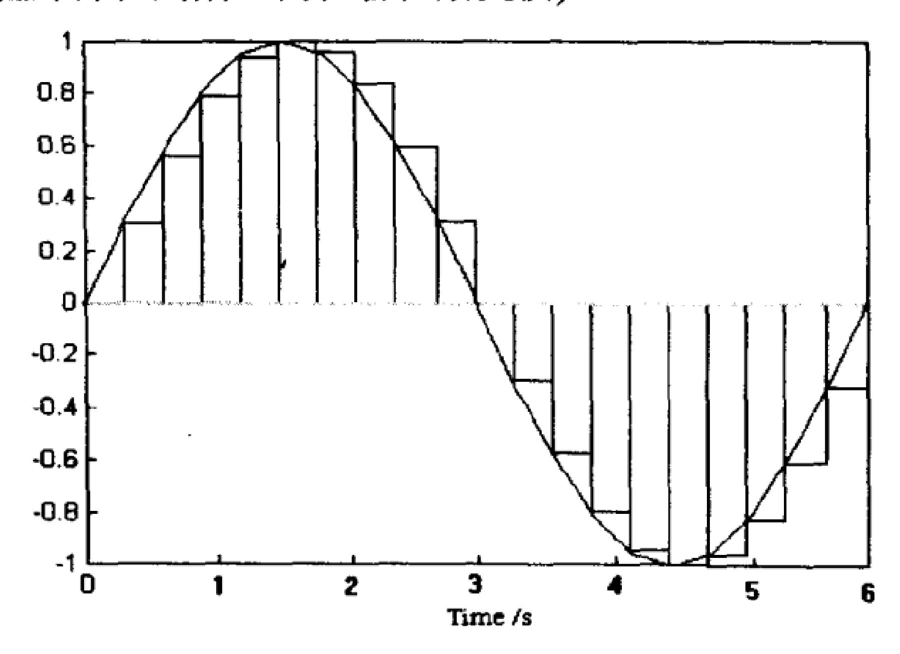


图 6-5 零阶保持器的输出波形

$$\frac{1-e^{-sT}}{s}$$

在开关输出和系统输出之间的总的传递函数是

$$\frac{1-e^{-sT}}{s}G(s)$$

无论什么时候，在装置前都有零阶保持器。因此对上面的式子取 z 变换如下：

$$Z\left\{\frac{1-e^{-sT}}{s}G(s)\right\}=Z\left\{\frac{G(s)}{s}\right\}-Z\left\{\frac{e^{-sT}G(s)}{s}\right\}$$

由此可以推出

$$G(z)_{\text{ZOH}}=(1-z^{-1})Z\left\{\frac{G(s)}{s}\right\}$$

为了得到 s 域内传递函数 $G(s)$的 z 变换，要完成下列的运算：由 $G(s)$得到 $g(t)$，用 nT 代替 t，采用 z 变换从而得到 $G(z)$。

我们用上述步骤对 $G(s)/s$ 进行 z 变换，以便得到零阶保持器的等效传递函数。

考虑如下的简单例子：

$$G(s)=\frac{1}{s+1}$$

$$G(z)_{\mathrm{ZOH}}=(1-z^{-1})Z\left\{\frac{1}{s(s+1)}\right\}$$

$$G(z)_{\mathrm{ZOH}}=\frac{(1-\mathrm{e}^{-T})}{z-\mathrm{e}^{-T}}$$

零阶保持器的 z 变换法也称为阶跃响应不变法，因为它使得离散系统与连续系统的阶跃响应相对应。到目前为止，它是在控制系统中最常用的转换技术，在附录 B 表 B-1 中给出了一些简单的零阶保持器等效式，这种离散化可以在状态空间中获得，零阶保持器状态空间矩阵如下：

$$\boldsymbol{A}_{\mathrm{d}}=\mathrm{e}^{\boldsymbol{A}T} \quad \boldsymbol{B}_{\mathrm{d}}=\int_0^T \mathrm{e}^{\boldsymbol{A}\sigma}\boldsymbol{B}\mathrm{d}\sigma \quad \boldsymbol{C}_{\mathrm{d}}=\boldsymbol{C} \quad \boldsymbol{D}_{\mathrm{d}}=\boldsymbol{D}$$

假如我们将矩阵按指数级数展开，完成积分，运用矩阵代数，离散矩阵 $\boldsymbol{B}$ 可由下式计算：

$$\boldsymbol{B}_{\mathrm{d}}=\boldsymbol{A}^{-1}[\boldsymbol{A}_{\mathrm{d}}-\boldsymbol{I}]\boldsymbol{B}=\mathrm{e}^{\begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}T}$$

最终的形式仅仅要求矩阵运算，零阶保持器是在 MATLAB 控制系统工具箱中的一个缺省的离散化技术。

6.6.3 差分变换法

为了将模拟滤波器转换成等效的数字滤波器，我们最关心的是保持原来传递函数的频率响应，要保持数字化传递函数的频率响应，最常用的方法是由积分和微分的近似得到，其方法有前向差分法、后向差分法和双线性变换法。用这些方法，可以对 $H(s)$ 中的每个 s 都用 z 的函数代替。

考虑下面的一种情况。我们想要知道当前的位置 $y(kT)$，已知 T 秒以前所在的位置，速度 v 在 T 时间内是恒定的，可以写出

$$v(kT)=\frac{y(kT)-y[(k-1)T]}{T}$$

对于速度 v，有几种选择，可以使用在 kT 秒时的速度、在 $(k-1)T$ 秒的速度，或者是两者的平均速度，第一种产生的近似被称作后向差分法。

$$v(kT)=\frac{y(kT)-y[(k-1)T]}{T} \tag{6-22}$$

对式(6-22)采用 z 变换得

$$V(z)=\frac{z-1}{Tz}Y(z) \tag{6-23}$$

在连续域内，速度被表示为 $sY(s)$，对应式(6-23)可以得出 s 和 z 之间的关系。

由后向差分法可求得

$$s=\frac{z-1}{Tz}$$

如果使用在$(k-1)T$ 秒时的速度，即采用前向差分法，则有

$$v[(k-1)T]=\frac{y(kT)-y[(k-1)T]}{T}$$

$$z^{-1}V(z)=\frac{1-z^{-1}}{T}Y(z)$$

由此可以得到

$$s=\frac{z-1}{T}$$

如果采用初始时和结束时的平均值，即采用双线性变换法，则有

$$s=\frac{2}{T}\frac{z-1}{z+1}$$

双线性变换法是将模拟滤波器转换成数字滤波器最常用的方法。

在状态空间的矩阵运算中，也可以采用这些变换技术。例如，在下面的例子中使用的后向差分法。在 s 域内的连续状态空间方程是

$$s\boldsymbol{X}(s)=\boldsymbol{A}\boldsymbol{X}(s)+\boldsymbol{B}\boldsymbol{U}(s)$$

$$Y(s) = CX(s) + DU(s)$$

用$(z-1)/Tz$代替s，应用z反变换得到接近的差分方程

$$x[(k+1)T] = \Psi x(kT) + \Psi TBu[(k+1)T]$$

式中，$\Psi = (I - TA)^{-1}$

为了得到状态空间响应，定义

$$\eta(kT) = x(kT) - \Psi TBu(kT)$$

根据新的状态 $\eta(kT)$，重写状态方程

$$\eta[(k+1)T] = \Psi\eta(kT) + \Psi^2 TBu(kT)$$

$$y(kT) = C\eta(kT) + (C\Psi TB + D)u(kT)$$

因此，离散状态空间矩阵是

$$A_d = \Psi \quad B_d = \Psi^2 TB \quad C_d = C \quad D_d = C\Psi TB + D$$

双线性变换矩阵和前向差分矩阵也可以被推导出来(见习题）。

比较这三种方法，有一些需要考虑的问题。首先，前向差分法不一定维持原系统的稳定性。例如，对于如下的模拟滤波器：

$$H(s) = \frac{1}{s+1}$$

其极点$p=-1$，肯定是稳定的，对其进行前向差分变换有

$$H(z) = \frac{T}{z-1+T}$$

其极点为$z=1-T$，假如T是大于 2 的数，则离散化后的极点在单位圆外，即数字滤波器的动态响应是不稳定的，当T小于 2 时，数字滤波器的极点在z平面的单位圆内，是稳定的。由此可见，由于采样周期长，会将一个稳定的模拟系统变换成不稳定的数字系统。

下面是三种变换方法的变换式。

前向差分法：　$z = 1 + sT$

后向差分法：　$z = 1/(1 - sT)$

双线性变换法：$z = (1 + sT/2)/(1 - sT/2)$

前向差分法实际上是泰勒级数指数$z = e^{sT}$展开式的前两项：

双线性变换法则是指数 $z = e^{sT}$ 的 Pada 近似。

s 平面的 jω 轴对应于 z 平面的单位圆。前向差分法和后向差分法都没有维持这种对应关系；而在双线性变换中，令 $s = j\omega$，则

$$z = \frac{1 + j\omega T/2}{1 - j\omega T/2}$$

从上述方程中，其幅值 $|z| = 1$，其相角为 $\theta_z = 2\tan\omega T/2$，当 ω 增加时，相角增加，而幅值保持在单位圆上不变。因此，在双线性变换中，维持了 jω 与 z 平面上的单位圆的对应关系，因而不会影响离散化后系统的稳定性。但是该变换会产生偏差问题，其离散化后动态响应会由于采样周期的变化而变化，从而有一定的偏差。当采样速度高时，偏差会减少，该问题将在后几节中介绍和讨论。

6.7 用于离散系统的 MATLAB 命令

在控制系统中的大部分的 MATLAB 命令，在数字控制系统中都有对应的命令。数字控制系统的命令格式通常以字母 d 起头，例如 dstep 、 dbode 、 dnyquist 等等，其句型格式与用法和在连续控制系统中几乎是相同的，不同的地方我们将详细地介绍。最重要的命令是 c2d 和 c2dm，其命令格式为

```
[Ad,Bd]=c2d(A,B,ts)
[Ad,Bd,Cd,Dd]=c2dm(A,B,C,D,ts,'method')
[numz,denz]=c2dm(num,den,ts,'method')
```

c2d 命令使用离散化的零阶保持器方法，它只有状态空间形式，而 c2dm 既有状态空间形式，又有传递函数形式，它可以用五种方法将连续系统转换成数字控制系统，其中参数 ts 是采样周期 t_s， method 选项允许用户采用五种转换方法之一，这五种转换方法是零阶保持器法、一阶保持器法、匹配法、双线性变换法、偏差的预补偿法。零阶保持器法和双线性变换法，已经在前面叙述过了，一阶保持器法是由零阶保持器法发展而来的，在这

种方法中，假设控制输入在采样周期内是以上个周期与本周期采样的平均速度线性变化的，而不象零阶保持器是恒定的。其它两种方法在后面讨论。假如没有指定的话，则认为是零阶保持器法。

求离散系统阶跃响应的命令格式为

[y,x]=dstep(A,B,C,D,ui,n)

[y,x]=dstep(num,den,n)

上述命令会给出一组阶跃响应的数据，若没有等式左边的参数，会自动地绘制出响应曲线。 ui 和 n 是可选项，对于多输入系统 ui 是用于指定哪个输入通道， n 是指采样数。和连续系统中的 step 命令有关的所有命令都可以在这里应用。其它时间响应命令是 dimpluse 、 dinitial 、 dlsim 。

离散系统时域响应的曲线是由 stairs 命令绘制的。这种命令产生梯形的图形曲线，它有如下的命令格式：

stairs(y) 或 stairs(x,y)

[xs,ys]=stairs(y) 或 [xs,ys]=stairs(x,y)

同样，没有等式左边的参数，会自动地绘制响应曲线。如果有两个参数 x 、 y ，将 y 绘制在 x 的特定点上，应该指定左边的参数，这样可以先不绘制曲线，再使用 plot 命令绘制出要求的梯形曲线。

离散系统的频域响应可以用 dbode 命令获得(同样可以使用 dnyquist 、 dnichols)。它有如下的命令格式：

[mag,phase]=dbode(A,B,C,D,ts,ui ,w)

[mag,phase]=dbode(num,den,ts,w)

式中， ts 是采样周期 t_s ，所有与连续系统中 bode 命令有关的命令在这里同样适用。

离散系统和连续系统的根轨迹是近似的，唯一的差别是坐标系不同。连续系统所用的是直角坐标，离散系统用的是极坐标。 zgrid 命令在单位圆内设置了阻尼系数和固有频率线，设置 “ hold ” 命令，以便在根轨迹上保持图形的重叠。其格式为

zgrid('new')

其它适用的命令有 ddcgain(用于计算直流增益)、 dsort(用幅值对一个矢量的复数特征值进行分类)、 drmode(建立一个随机而稳定的模型)、 ddamp 命令。 ddamp 命令有如下的格式：

ddamp(A,ts)

该命令可以得到矩阵 A 的特征值(假如 A 是个矢量，则得到多项式的根)、幅值、在 s 平面的等效阻尼率和固有频率。

下面的例子给出了求取连续系统模型的单位阶跃响应和它离散化后零阶保持器模式的阶跃响应的方法。

【例 6-1 】求取零阶保持器传递函数的阶跃响应。

使用零阶保持器将 $G(s)$转换为 $G(z)$，采样周期 T=0.1s ，比较 $G(s)$和 $G(z)$的阶跃响应。

$$G(s)=\frac{10}{(s+2)(s+5)}$$

将连续系统的传递函数用下面零阶保持器法转换成离散系统传递函数。

```
num=10;den=[1,7,10];ts=0.1;
[n_zoh,d_zoh]=c2dm(num,den,ts)
```

上述程序可以给出如下结果：

```
n_zoh =
        0    0.0398    0.0315
d_zoh =
     1.0000   -1.4253    0.4966
```

下面求取 $G(s)$和 $G(z)$的阶跃响应，首先将 $G(s)$离散化，然后用 step 命令求取连续系统的阶跃响应，用 dstep 命令求取离散系统的阶跃响应，并用 hold 命令将其画在同一坐标系内。MATLAB 程序 6-1 是求取 $G(s)$和 $G(z)$阶跃响应的程序清单。

```
% MATLAB PROGRAM  6-1
nc=10;dc=[1,7,10];ts=0.1;i=[0:35];time=i*ts;
[n_zoh,d_zoh]=c2dm(nc,dc,ts);
yc=step(nc,dc,time);
```

```
y_zoh=dstep(n_zoh,d_zoh,36);
[xx,yy]=stairs(time,y_zoh);
plot(time,yc,'r'),hold
plot(xx,yy,'r'),hold;grid
end
```

图 6-6 显示连续系统和用零阶保持器法离散化后数字系统的阶跃响应是相同的。因此，这种数字化方法总是产生与连续系统相同的阶跃响应。

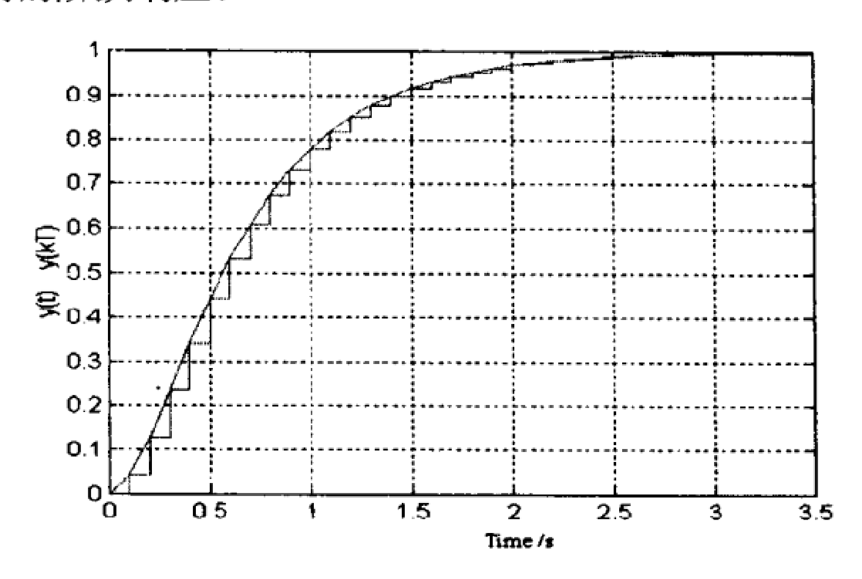


图 6-6 连续系统和用零阶保持器法离散化后的数字系统的阶跃响应

下面是一个比较连续系统和零阶保持器等价系统的频率响应的例子。我们将看到，离散系统的频率响应与连续系统的频率响应的一致性，很大程度上取决于采样周期的选取。

【例 6-2】零阶保持器法的频率响应。

用零阶保持器法将 $H(s)=10/(s+1)$离散化，采样周期 T 分别取 1,0.5,0.15，比较连续系统和三个离散传递函数的频率响应。

MATLAB 程序 6-2 是求取连续系统和三个离散系统频率响应的程序清单：

```
% MATLAB PROGRAM 6-2
nc=10;dc=[1,1];
w=[0:0.2:10];
```

```
mc=bode(nc,dc,w);
ts1=1;ts2=0.5;ts3=0.1;
[nz1,dz1]=c2dm(nc,dc,ts1);
mz1=dbode(nz1,dz1,ts1,w);
[nz2,dz2]=c2dm(nc,dc,ts2);
mz2=dbode(nz2,dz2,ts2,w);
[nz3,dz3]=c2dm(nc,dc,ts3);
mz3=dbode(nz3,dz3,ts3,w);
plot(w,mc,'g')
hold
plot(w,mz1,'y')
plot(w,mz2,'b')
plot(w,mz3,'r')
hold
xlabel('ω  / (rad/s)')
ylabel('gain  / dB')
end
```

其频率响应曲线见图 6-7 。

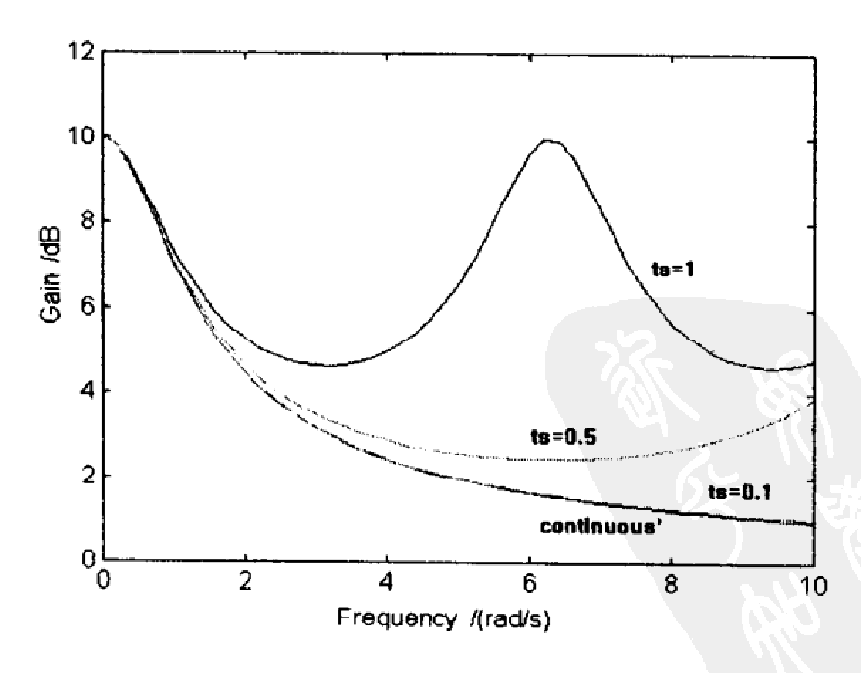


图 6-7　频率响应曲线

从图 6-7 可以看到一些相关的特性。首先，ZOH 的频率响应取决于采样周期，当采样周期减小到 $t_s=0.1s$ 时，离散化系统的频率响应接近于连续系统的频率响应；当采样周期较大时，频率响应几乎是周期性的。

由香农定理可知，采样频率大于奈奎斯特频率的重要性。对于 $H(s)=10/(s+1)$的传递函数，频带宽度为 1.0rad/s，因此，采样频率至少是 2rad/s，换句话说，采样周期应该是

$$T_{\max}=\frac{2\pi}{\omega_s}=\frac{2\pi}{2}=3.14s$$

在上面的分析中，最低采样频率 T=1s 是奈奎斯特频率的 3 倍，响应仍然不能与原连续系统相一致。为了获得与原连续系统相同的频率响应，需要把频率提高到奈奎斯特频率的 30 倍。

因此，零阶保持器法不能保持离散化后的系统的频率响应与原连续系统的频率响应一致。

数字系统传递函数的频率响应，有周期性振荡的特性，这不是由于所使用的不同的数字化技术产生的，在所有的 z 变换传递函数中，它是普遍存在的。

为了认证这一点，看一下简单的传递函数

$$H(z)=\frac{1}{z-0.5}$$

由于 $z=e^{sT}\,(s=j\omega)$，可以从数字系统传递函数

$$H_z(\omega)=\frac{1}{e^{j\omega T}-0.5}$$

得到频率响应。

然而，当 $\omega T=0, 2\pi, 4\pi, \cdots$ 时，函数 $e^{j\omega T}$ 不断地重复，因此，在频率区间内，传递函数将不断地重复，即所有数字系统的伯德图都是周期性的，在采样频率 $2\pi/T$ 点重复，随着 T 的减小，其重复的频率区间增大。没有真正的低通数字滤波器，只是在其频率值远远小于采样频率时，才具有较好的低通性能。

什么时候使用零阶保持器法呢？通常用这种方法将一个连

续系统装置数字化，这是因为我们常常是对装置的时域响应感兴趣，而很少用零阶保持器法将一个连续系统的滤波器变换成数字滤波器，因为我们已经看到，这种变换技术不能真实地再现它原来的频率响应，而设计滤波器就是为了改善它的频率特性。

6.8 偏差问题

考虑一频带宽度为 a 的模拟滤波器

$$H(s)=\frac{1}{s/a+1} \tag{6-24}$$

该滤波器的双线性变换为

$$H(z)=\frac{1}{\frac{2}{T}\cdot\frac{z-1}{z+1}\cdot\frac{1}{a}+1}$$

为了求取数字滤波器的频率响应，令 $z=\mathrm{e}^{\mathrm{j}\omega T}$，则

$$\frac{z-1}{z+1}\rightarrow\frac{\mathrm{e}^{\mathrm{j}\omega T}-1}{\mathrm{e}^{\mathrm{j}\omega T}+1}$$

从上式的分子、分母提取因子 $\mathrm{e}^{\mathrm{j}\omega T/2}$ 则

$$\frac{\mathrm{e}^{\mathrm{j}\omega T}-1}{\mathrm{e}^{\mathrm{j}\omega T}+1}\rightarrow\frac{\mathrm{e}^{\mathrm{j}\omega T/2}(\mathrm{e}^{\mathrm{j}\omega T/2}-\mathrm{e}^{-\mathrm{j}\omega T/2})}{\mathrm{e}^{\mathrm{j}\omega T/2}(\mathrm{e}^{\mathrm{j}\omega T/2}+\mathrm{e}^{-\mathrm{j}\omega T/2})}\rightarrow\frac{2\mathrm{j}\sin\omega T/2}{2\cos\omega T/2}=\mathrm{j}\tan\frac{\omega T}{2}$$

由于 $s=\mathrm{j}\omega$，则数字滤波器变为

$$H_z(\omega)=\frac{1}{\mathrm{j}\left(\frac{2}{T}\tan\frac{\omega T}{2}\right)\cdot\frac{1}{a}+1} \tag{6-25}$$

由于一阶滤波器的穿越频率是在虚数项等于实数项的点，因此，数字滤波器的带宽是

$$\omega_{\mathrm{BW}}=\frac{2}{T}\arctan\left(\frac{aT}{2}\right) \tag{6-26}$$

该公式给出了连续系统和数字系统等效带宽之间的关系，也称之为偏差公式。它显示了采用双线性变换频带宽度是怎样产生偏差的，采样越快，则采样周期 T 越小，由于一个数越小，它的反正切越接近它。可以推导出，对于很高的采样频率，数字滤波

器与模拟滤波器有相同的带宽。然而，在控制系统中，常常使用较低的采样频率，因此离散化的数字滤波器与原来的模拟滤波器的带宽有着相当大的区别。

6.8.1 偏差的预补偿方法

在对连续系统数字化以前，对模拟滤波器进行偏差的预补偿。在 $H(s)$中，用 $\overline{a}$ 代替 a，$\overline{a}$ 被定义为

$$\overline{a}=\frac{2}{T}\tan\frac{aT}{2} \tag{6-27}$$

假如我们用

$$H_{\mathrm{pw}}(s)=\frac{1}{s/\overline{a}+1} \tag{6-28}$$

代替原来的模拟滤波器，应用双线性变换法，将其数字化，使数字滤波器与模拟滤波器有相同的带宽。

假如使用偏差预补偿方法，使 $H(s)$变为 $H_{\mathrm{pw}}(s)$ 如下

$$H(s)=\frac{K}{s+a} \quad H_{\mathrm{pw}}(s)=\frac{K}{s+\overline{a}} \tag{6-29}$$

注意，在上面的变换中，已经改变了其低频增益，为了不改变低频增益，首先用 a 除上式，然后再用偏差的预先补偿方法，得到下式所示结果。

$$H(s)=\frac{K/a}{s/a+1} \Rightarrow H_{\mathrm{pw}}(s)=\frac{K/\overline{a}}{s/\overline{a}+1} \tag{6-30}$$

【例 6-3】 用偏差的预补偿的双线性变换法。

采样周期分别为 T=0.5s 、 T=1.0s ，用双线性变换将 $H(s)$ 数字化。

$$H(s)=\frac{10}{s+2}$$

首先用 2 除分子分母，从而变成其标准型

$$H(s)=\frac{5}{s/2+1}$$

对于 T=0.5s ，则

$$\bar{a}=\frac{2}{0.5}\tan\frac{2\times 0.5}{2}=2.1852$$

对于双线性变换，将用 c2dm 命令得到

$$H_1(s)=\frac{5}{s/2.1852+1} \quad \rightarrow \quad H_1(z)=1.7665\frac{z+1}{z-0.2934}$$

上面的步骤是用于说明这种变换的，使用偏差补偿的 c2dm 命令，简化了这种过程。MATLAB 程序 6-3 是采用偏差补偿法求取离散化传递函数的程序清单。

```
% MATLAB PROGRAM 6-3
nc=10;dc=[1,2];
ts1=0.5;ts2=1;
[nz1,dz1]=c2dm(nc,dc,ts1,'p',2);
[nz2,dz2]=c2dm(nc,dc,ts2,'p',2);
printsys(nz1,dz1,'z')
printsys(nz2,dz2,'z')
end
```

结果是

```
num/den =
              1.766 z+ 1.766
              --------------
                z- 0.2934
num/den =
              3.045 z+ 3.045
              --------------
                z+ 0.218
```

注意带'z'选择项的 printsys 命令的用法是用于表示求取的是离散传递函数，方法项的第一个字母的意义是补偿方法(如'p'是指偏差的预补偿方法)。MATLAB 程序 6-4 给出绘制频域响应曲线的程序清单。

```
% MATLAB PROGRAM 6-4
nc=10;dc=[1,2];
w=[0.01:0.2:12];
ts1=0.5;ts2=1;
```

```
[nz1,dz1]=c2dm(nc,dc,ts1,'p',2);
[nz2,dz2]=c2dm(nc,dc,ts2,'p',2);
[mc,phc]=bode(nc,dc,w);
[mz1,phz1]=dbode(nz1,dz1,ts1,w);
[mz2,phz2]=dbode(nz2,dz2,ts2,w);
semilogx(w,mc,'r-')
hold
semilogx(w,mz1,'g-.')
semilogx(w,mz2,'r--')
xlabel('ω / (rad/s)')
axis([0 12 -1 6]);
end
```

图6-8给出了连续系统$H(s)$的伯德图和两个预补偿传递函数的数字化系统的伯德图。正如所估计的，所有的三根曲线都有着相同的带宽。然而三个传递函数的频率响应却是不同的。

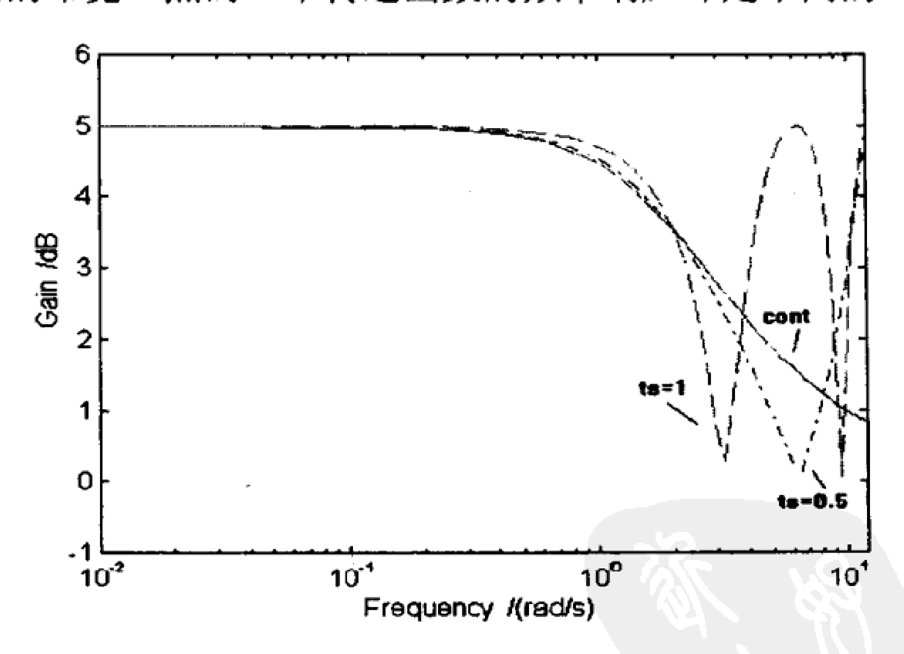


图 6-8 使用偏差预补偿的双线性变换的频率响应

当对较高阶的传递函数进行数字化时，推导偏差预补偿的公式不是那么容易的。预补偿的每一个零极点是什么？是要保持带通滤波器的中央频率不变，还是保持带宽不变？还是两者都不

变？我们只能使离散传递函数的数字化响应在某一频率上与原来的连续传递函数的响应保持一致。

最好的方法是直接在 z 平面上设计滤波器或补偿器，在下一节将介绍这种方法。然而，在控制系统中还是经常会面临用数字化的等效滤波器代替模拟滤波器的任务。下面的改进提供了一种较为合理的方法，该方法称之为临界频率预补偿方法。

6.8.2 临界频率的预补偿方法

重新看一下双线性变换的近似表达式

$$s = K\frac{z-1}{z+1} \tag{6-31}$$

式中，$K=2/T$。这就提出如下问题：能否找到某一个 K 值，以便在某一个特定的频率上使上述近似的关系变成精确的等式，所选的这个频率为滤波器的临界频率，即 $\omega=\omega_c$，令 $s=j\omega_c$

$$j\omega_c = jK\tan\frac{\omega_c T}{2}$$

假如选取

$$K=\frac{\omega_c}{\tan(\omega_c T/2)}$$

则 $H(s)$和 $H(z)$在 $\omega=\omega_c$ 点将是相等的。因此，采用临界频率预补偿技术有

$$s \to \frac{\omega_c}{\tan(\omega_c T/2)}\cdot\frac{z-1}{z+1}$$

当 T 变小时，上述转换接近于双线性变换公式。使用偏差预补偿方法的 c2dm 命令可以完成这种变换。例 6-4 给出了对陷波滤波器进行数字化时，采用临界频率预补偿方法的例子。

【例 6-4】临界频率预补偿方法。

对于采样频率 T=0.1s，对下列陷波滤波器进行数字化：

$$H(s)=\frac{s^2+0.2s+100}{s^2+10s+100}$$

我们注意到上式的陷波发生在 10rad/s 点，这点就是临界频

率，分别画出模拟滤波器、没有预补偿的数字陷波滤波器和带补偿的数字陷波滤波器的幅频伯德图。

在临界频率点的偏差预补偿产生如下的滤波器：

$$\frac{n(z)}{d(z)}=\frac{0.7098z^2-0.7606z+0.6979}{z^2-0.7606z+0.4077}$$

没有偏差预补偿的数字陷波滤波器如下：

$$\frac{n(z)}{d(z)}=\frac{0.7200z^2-0.8571z+0.7086}{z^2-0.8571z+0.4286}$$

MATLAB 程序 6-5 是设计滤波器及产生其频率响应的程序清单。

```
% MATLAB PROGRAM 6-5
nc=[1,0.2,100];dc=[1,10,100];
w=[1.5:0.2:20];ts1=0.1;
[nz1,dz1]=c2dm(nc,dc,ts1,'p',10);
[nz2,dz2]=c2dm(nc,dc,ts1);
[mc,phc]=bode(nc,dc,w);
[mz1,phz1]=dbode(nz1,dz1,ts1,w);
[mz2,phz2]=dbode(nz2,dz2,ts1,w);
semilogx(w,mc,'k-')
hold
semilogx(w,mc,'y-')
semilogx(w,mz1,'r--')
semilogx(w,mz2,'g-.')
xlabel('ω / (rad/s)')
ylabel('margin / dB')
axis([1.5 20 -0.2 1.5]);
end
```

带偏差补偿与不带偏差补偿的数字陷波滤波器的伯德图见图 6-9 。

图 6-9 给出了模拟滤波器和数字滤波器的幅频特性，使用临界频率预补偿技术，可以保持原模拟滤波器的陷波频率，即只能

使其在某一点频率及其附近有相同的频率响应。然而，使用该方法或其它数字化方法，不能控制频带宽度或其它参数。

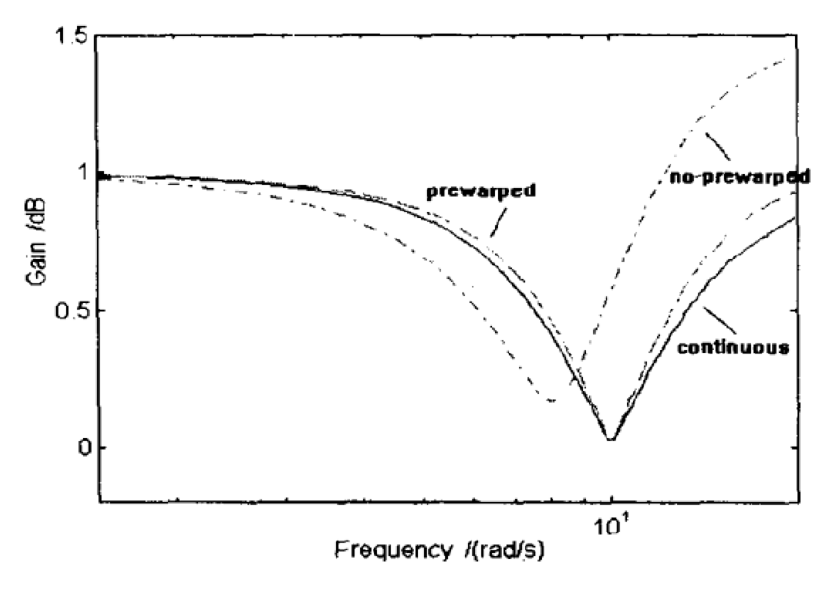


图 6-9 数字化系统的伯德图

6.9 数字补偿器

设计数字补偿器有如下一些方法：

(1) 将按经典方法设计的模拟补偿器转化为等效的数字补偿器。

(2) 用频率转换方法直接设计数字补偿器。

(3) 用状态空间方法直接设计数字补偿器。

正如前面几节所知道的，必须特别注意第一种方法，将模拟补偿器转换为数字补偿器，一般采用双线性变换方法(带偏差补偿或不带偏差补偿)，在较高的采样频率时，这些方法是非常有效的。然而，在采样频率较低时，这种转换经常会带来不可预料的问题。这是因为数字补偿器与转换前的模拟补偿器，其频率响应有很大的区别。

在 z 平面上的设计，常允许设计者不考虑采样周期的影响，在这方面，最有效的方法是根轨迹法。我们已经看到 z 域内数字补偿器的频率响应曲线没有模拟补偿器的那么好。后面将介绍 ω 变换技术，该技术允许使用伯德图设计数字补偿器。

6.9.1 PID 控制

在数字系统中， PID 控制是一种非常有效的方法。事实上，PID 控制器在工业过程控制中是一种经济实用的控制器。积分环节增加了系统的型，而减少了系统的稳态误差，微分环节增加了阻尼，而使得系统稳定。最常见的离散 PID 控制器有如下的形式：

$$K_{\mathrm{P}}+K_{\mathrm{D}}\frac{z-1}{z+1}+K_{\mathrm{I}}\frac{zT}{z-1}$$

在上列公式中，速度是本次采样值与上次采样值之差除以采样周期，即 $(1-z^{-1})/T$ 。积分项则是在采样周期内某一值下的面积。

在前几章所描述的 Ziegler-Nichols 方法也适用于数字系统，不要忘记，在公式里包含着采样周期。

首先，令 $K_{\mathrm{D}}=K_{\mathrm{I}}=0$，增加比例放大倍数 K_{P}，直到系统振荡(即使系统的闭环极点位于 z 平面的单位圆上)，此时的 $K_{\mathrm{P}}=K_{\mathrm{m}}$。比例系数乘以 0.6，其它的两个增益可计算如下：

$$K_{\mathrm{P}}=0.6K_{\mathrm{m}} \quad K_{\mathrm{D}}=\frac{K_{\mathrm{P}}\pi}{4\omega_{\mathrm{m}}} \quad K_{\mathrm{I}}=\frac{K_{\mathrm{P}}\omega_{\mathrm{m}}}{\pi}$$

式中，ω_{m} 是系统的振荡频率。振荡频率可以由极点位于单位圆上的角度得到，$\omega_{\mathrm{m}}=\theta/T$，振荡频率也可以由运行闭环系统的阶跃响应使 $K_{\mathrm{P}}=K_{\mathrm{m}}$ 时得到。

注意到上述方法没有任何的限制条件，因此这种设计方法设计出的过程控制器具有很好的性能。

【例 6-5】 用 Ziegler-Nichols 方法设计 PID 控制器。

系统传递函数为 $G(s)=10/[s(s+2)]$,采样周期 T=0.25s，设计一个 PID 控制器。

其设计步骤：

用零阶保持器将其离散化，画出 $G(z)$的根轨迹图，决定 K_{m} 和 K_{D}，计算 PID 控制器的参数，画出阶跃响应。 MATLAB 程序 6-6 是设计 PID 控制器的程序清单。

```
% MATLAB PROGRAM 6-6
```

```
nc=10;dc=[1,2,0];ts=0.25;
[nz1,dz1]=c2dm(nc,dc,ts);
axis('square'),zgrid('new'),
rlocus(nz1,dz1)
[km,pole]=rlocfind(nz1,dz1);
wm=angle(pole(1))/ts;
kp=km*0.6,kd=kp*pi/(4*wm),ki=kp*wm/pi
end
```

由上述程序得到的 PID 控制器参数如下:

```
selected_point =
    0.5703 + 0.8206i
kp =
    1.0468
kd =
    0.2134
ki =
    1.2836
```

然后用 series 和 cloop 命令求取开环和闭环系统传递函数。

用 Ziegler-Nichols 方法设计的根轨迹见图 6-10 。

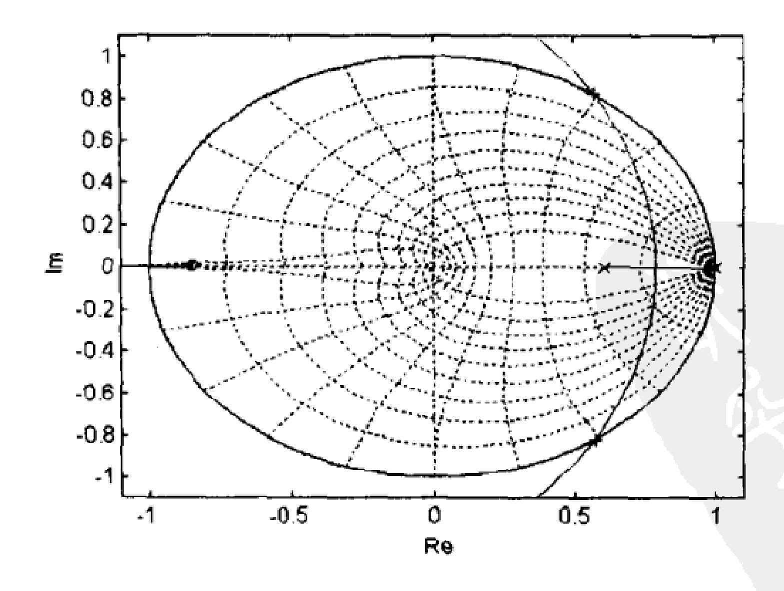


图 6-10　没有 PID 控制器的系统根轨迹图

加入前面所设计的 PID 控制器后，系统根轨迹图见图 6-11 。

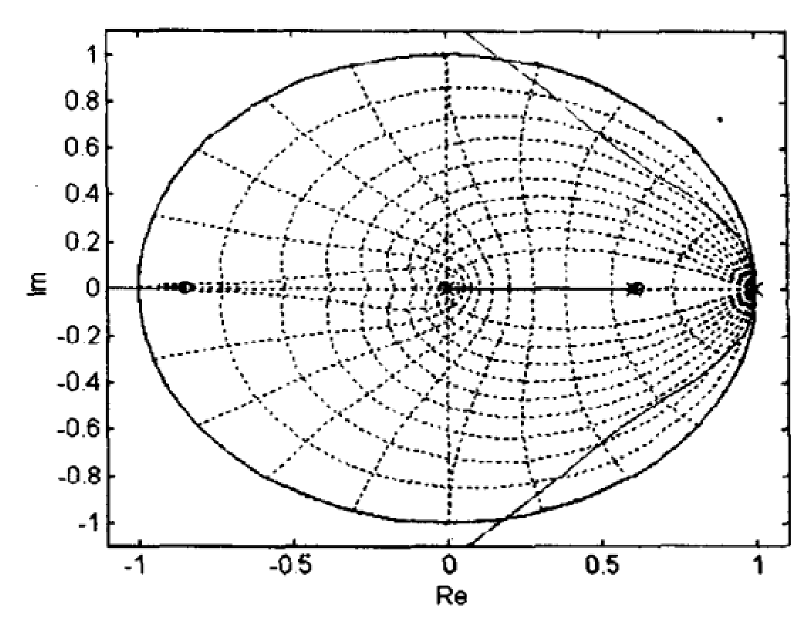


图 6-11　有 PID 控制器的系统根轨迹图

加入 PID 控制器后，系统的阶跃响应曲线见图 6-12 。

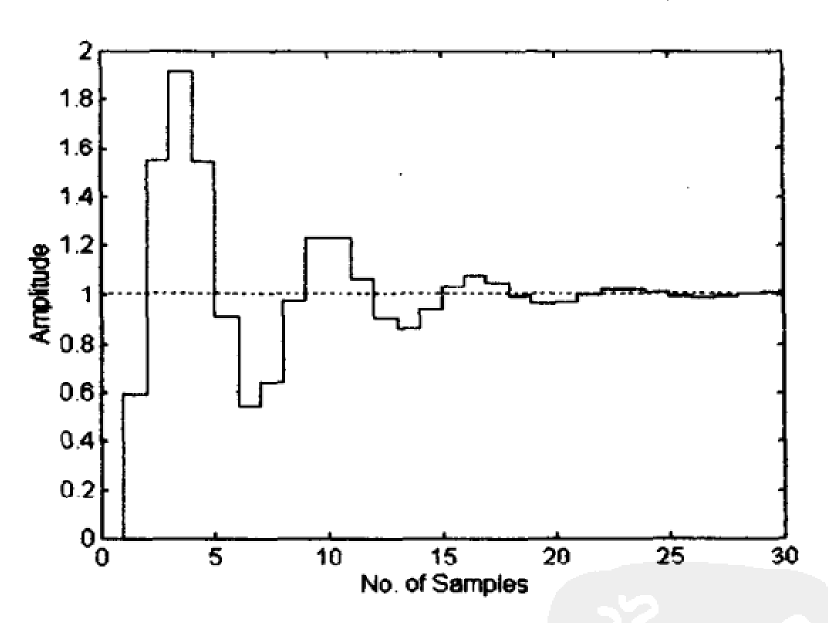


图 6-12　加入 PID 控制器后系统的阶跃响应曲线

MATLAB 程序 6-7 为求根轨迹图及系统阶跃响应的程序清单：

```
% MATLAB PROGRAM 6-7
nc=10;dc=[1,2,0];ts=0.25;
[nz1,dz1]=c2dm(nc,dc,ts);
```

```
axis('square'),zgrid('new'),
rlocus(nz1,dz1)
[km,pole]=rlocfind(nz1,dz1);
wm=angle(pole(1))/ts;
kp=km*0.6;
kd=kp*pi/(4*wm);
ki=kp*wm/pi;
nz2=[kp*ts+kd+ki*ts*ts -(kp*ts+2*kd) kd];
dz2=[ts -ts 0];
[nzk,dzk]=series(nz2,dz2,nz1,dz1);
axis([-1.1 1.1 -1.1 1.1]);
figure;
dstep(nzk,dzk)
axis('square'),
zgrid('new'),
rlocus(nzk,dzk)
[nt,dt]=cloop(nzk,dzk,-1);
axis([-1.1 1.1 -1.1 1.1]);
figure;
dstep(nt,dt)
end
```

系统的特征值等参数如下：

特征值	幅值	等效阻尼比	等效频率
0.4745+0.6818i	0.8306	0.1893	3.9221
0.4745-0.6818i	0.8306	0.1893	3.9221
0.6043	0.6043	1.0000	2.0146
0.4617	0.4617	1.0000	3.0916

由系统的阶跃响应曲线及特征值等参数可知，系统有小的阻尼率，有 90 %的超调量，阶跃响应在 20 个采样周期时达到稳定，即稳定时间约为 5s(稳定时间=采样数×采样周期)。在补偿后的

根轨迹上使用 rlocfind 命令，可以求得系统在增益为 3.4 时变得不稳定了，幅频裕量为 10dB 。

6.9.2 PID 控制器分析技术

第 4 章讨论的分析技术是由相角增益特性计算出来的，应用在这里不容易计算。我们可以推导出一种建立在根轨迹基础上的新方法。

首先，由要求的稳定误差决定系数 K_I，然后，由时域定义 z 平面上所期望的 z_1 点，如果 z_1 是在根轨迹上，则

$$\left(K_P + K_D\frac{z_1-1}{z_1T} + K_I\frac{z_1T}{z_1-1}\right)G(z_1) = -1 \tag{6-32}$$

由于 K_I 和 z_1 是已知的，则

$$K_P + K_D\frac{z_1-1}{z_1T} = -\frac{1}{G(z_1)} - K_I\frac{z_1T}{z_1-1}$$

上面的方程是复数方程，方程两边的虚部和实部分别相等，可以得到两个方程，令 R 和 X 分别为方程右边的实部和虚部，并定义

$$\alpha + \mathrm{j}\beta = \frac{z_1-1}{z_1T}$$

由此可以得到

$$K_P + K_D(\alpha + \mathrm{j}\beta) = R + \mathrm{j}X$$

因此

$$K_D = X/R \quad K_P = R - \alpha K_D \tag{6-33}$$

【例 6-6】 PID 控制器的分析设计。

系统的传递函数为

$$G(s) = \frac{10}{s(s+2)}$$

PID 控制器应满足下列要求：

T=0.25s ，ζ=0.707 ，$\omega_n = 1.414\,\mathrm{rad/s}$，稳态误差在斜坡输入情况下为 0 。

期望的 s 平面上的极点是 $s_1 = -1 \pm \mathrm{j}$， z 平面极点的位置是

$z_1 = e^{s_1 T} = 0.7546 \pm j0.1927$。

由于系统的传递函数是典型 I 型系统， PID 控制器将增加系统的阶数而成为典型II型系统，因此，这样已经满足稳态误差的要求，选取 K_I 是不受限制的。初选 $K_I = 4$ ，用本章附录 A 的程序清单 1 求取 K_P 和 K_D 。

由以上方法求得的闭环系统有一个极点的幅值是 1.5 ，由于在单位圆以外，因此，系统是不稳定的。假如比较闭环极点和期望极点的位置，程序已经在期望的位置放置了一个极点。这是该分析方法存在的问题。用上述方法，可以保证系统的一个极点是在期望的位置上，但是不能控制其它极点的位置。

假如再运行一次上述仿真，同时令 $K_I = 1$ ，就可以获得稳定的控制系统，其系统的阶跃响应见图 6-13 ，有 PID 控制器的系统的根轨迹图见图 6-14 ，各参数为 R=0.55 、X=0.51 、α=-0.97 、β=1.27 ， PID 控制器参数是 K_P =0.95 ， K_D =0.40 ， K_I =1.0 。

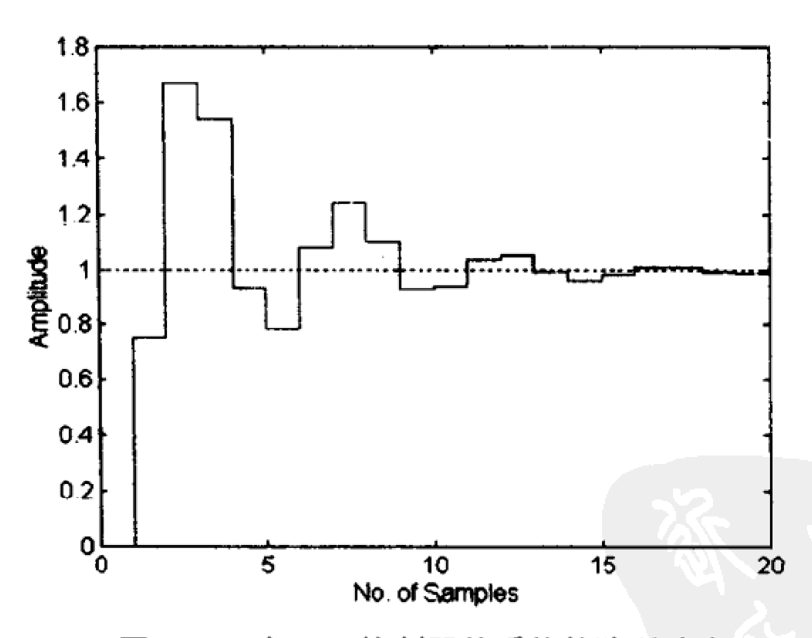


图 6-13　有 PID 控制器的系统的阶跃响应

ddamp 命令可以提供带 PID 控制器的系统的下列信息：

特征值	幅值	等效阻尼比	等效频率
0.7546+0.1927i	0.7788	0.7071	1.4142

0.7546-0.1927i	0.7788	0.7071	1.4142
0.1714+0.7593i	0.7784	0.1826	5.4871
0.1714-0.7593i	0.7784	0.1826	5.4871

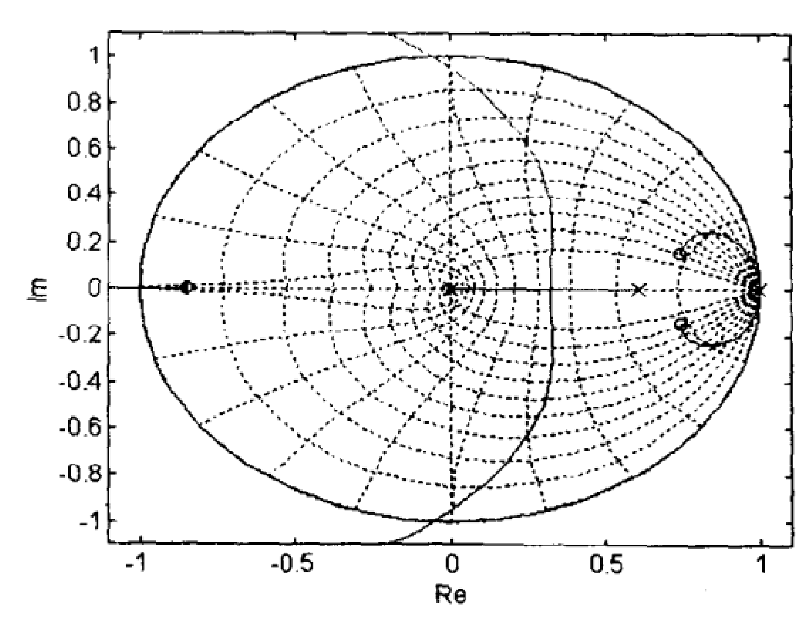


图 6-14　有 PID 控制器的系统的根轨迹图

极点的幅值是 0.7784 和 0.7788，验证了这是一个稳定系统，设计的系统其阻尼率和固有频率是满足要求的，注意其阶跃响应有着很大的超调量，这主要是因为这种方法不能控制所有零极点的位置。

6.9.3　超前-滞后补偿

离散系统的超前-滞后补偿器，可以用根轨迹设计。这是由于离散系统的根轨迹与连续系统的根轨迹的数字方法是相同的。在前面所介绍的有关的绘图方法，在这里是同样适用的。所不同的是，离散系统是在 z 平面的单位圆里，而连续系统则是在 s 平面的左半平面里。

由于被限制工作在单位圆里，因此产生一些问题，在较低增益的情况下，由稳定的连续系统离散化后得到的离散系统是不稳定的。离散系统传递函数常常比原来的连续系统传递函数有更多的零点，这会使系统的根轨迹很复杂，而所有的根轨迹都挤压在一个很小的单位圆的区域内。

为了有助于超前-滞后补偿器的设计，我们已经研制了一种分析方法，在设计时，总是最先考虑采用这种方法。记住，PID 分析设计的问题，可以保证期望极点在所希望的位置上，然而，却无法保证这些是主导极点，最终的数字补偿器的形式是

$$K(z)=K_{\mathrm{c}}\frac{z+a}{z+b} \tag{6-34}$$

由于可以求得在 $z=1$ 时的稳态误差常数，因此把补偿器的传递函数变为

$$K(z)=\overline{K}\frac{(z-1)/v+1}{(z-1)/w+1}=\overline{K}\frac{\bar{z}/v+1}{\bar{z}/w+1}$$

式中，$\bar{z}=z-1$, $a=v-1$, $b=w-1$, $K_{\mathrm{c}}=\overline{K}w/v$, 则当 $z=1$ 时，$\bar{z}=0$。$\overline{K}$ 是补偿器的稳态误差系数。

由系统的动态性能要求，可以决定在 s 平面的期望极点 s_1，令 $z_1=\mathrm{e}^{s_1T}$，得到在 z 平面的位置 z_1 和 $\bar{z}_1$，其推导过程如下：因为 z_1 是在根轨迹上

$$K(z_1)G(z_1)=\overline{K}\frac{\bar{z}_1/v+1}{\bar{z}_1/w+1}G(z_1)=-1$$

$$\frac{\bar{z}_1/v+1}{\bar{z}_1/w+1}=\frac{-1}{\overline{K}G(z_1)}$$

由此可以推导出 $1/v$ 为

$$v^{-1}=-\frac{w^{-1}}{\overline{K}G(z_1)}-\frac{1}{\bar{z}_1}\left(1+\frac{1}{\overline{K}G(z_1)}\right)$$

由于补偿器的零极点一定是实数，v^{-1}、w^{-1}，也一定是实数，因此方程两边的虚部一定为 0，令方程左边虚部为 0，导出 w^{-1} 为

$$w^{-1}=\frac{-\mathrm{Im}\left[1/\bar{z}_1+1/(\bar{z}_1\overline{K}G(z_1))\right]}{\mathrm{Im}\left[1/(\overline{K}G(z_1))\right]} \tag{6-35}$$

$$v^{-1}=-\frac{w^{-1}}{\overline{K}G(z_1)}-\frac{1}{\bar{z}_1\overline{K}G(z_1)}-\frac{1}{\bar{z}_1} \tag{6-36}$$

设计过程见本章附录 A 中程序清单 2 。

【例 6-7】 数字超前补偿器的设计。

系统传递函数为

$$G(s)=\frac{400}{s(s^2+30s+200)}$$

采样周期为 0.05s ，要求系统的稳态误差为 20 %， $\zeta=0.5$ ，$\omega_n=14\text{rad/s}$ 。

由稳态误差为 20 %，要求误差常数为 5 ，由于系统自身有一个误差常数 2 ，因此补偿器的误差系数应为 2.5 ，即 $\overline{K}=2.5$ 。

动态性能指标要求 s 平面的根为$-7+\text{j}12.12$ ，这可由前面几章介绍的方法求得，系统的调节时间 $t_s=0.7\text{s}$ ， z 平面的期望极点在 $0.5792+\text{j}0.4014$ ，用本章附录 A 中的程序清单 2 可以得到补偿器的传递函数为

$$K(z)=9.9935\frac{z-0.7609}{z-0.0441}$$

MATLAB 的 ddamp 命令可以得到如下一些信息：

特征值	幅值	等效阻尼系数	等效频率
Eigenvalue	Magnitude	Eq. Damping	Eq. Freq. (rad/s)
0.8268+0.0000i	0.8268	1.0000	3.8049
0.5816-0.3979i	0.7047	0.5039	13.8924
0.5816+0.3979i	0.7047	0.5039	13.8924
-0.0228-0.0000i	0.0228	0.7692	98.3196

闭环零点在：

-2.6170　0.7609　-0.1806

补偿后系统的根轨迹见图 6-15 。

所有的闭环极点都在单位圆内，但是在单位圆外有一个零点，图 6-16 给出了系统的阶跃响应曲线。

我们已经得到了期望的复合极点和调节时间 $t_s=1.0\text{s}$ 。阶跃响应与有主导极点的二阶系统有较大的差别，正如从上面给出的零极点的构成可以看到，在这种情况下，复合极点不是主导极点。

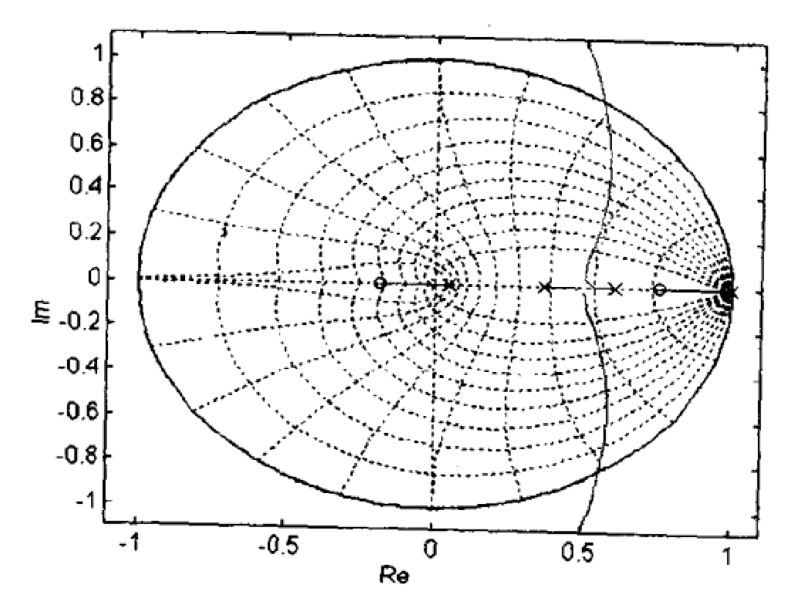


图 6-15　超前设计的根轨迹图

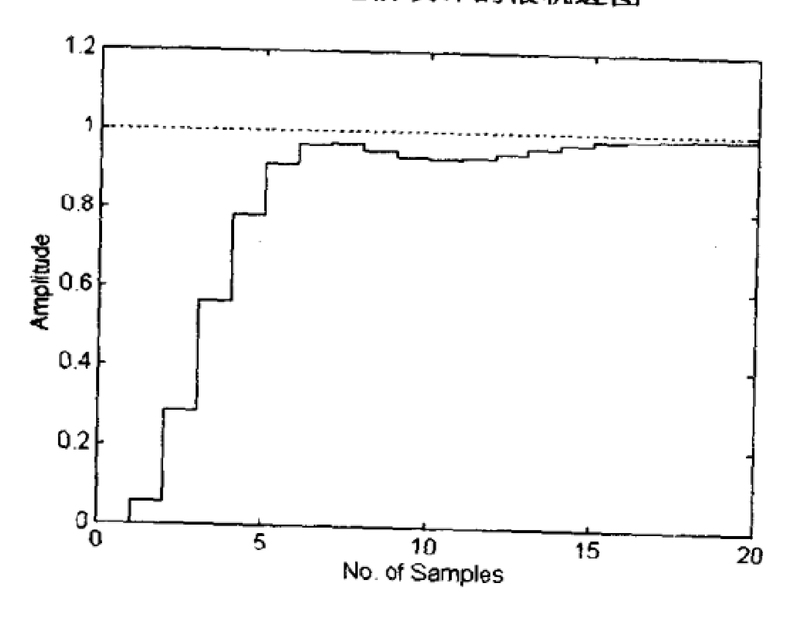


图 6-16　超前校正的阶跃响应曲线

6.9.4　ω变换

前面已经讨论过两种设计数字补偿器的方法。用各种变换技术将模拟补偿器变换成数字补偿器；在 z 平面上直接设计数字补偿器。当采样频率很高时，可以采用前一种变换方法。而在采样频率较低时，则最好采用 z 平面上直接设计补偿器的方法。

z 平面的设计方法有一些缺点。其一，所有的设计必须被限定在单位圆的区域内。其二，幅频特性穿越频率和相角裕量的伯

德图设计方法并不能应用到 z 平面上。该方法可以应用在连续系统的传递函数中，其传递函数的稳定区域是闭环极点在左半平面内。

ω变换可以将 s 域内的所有设计方法应用于数字系统的设计中，在 ω变换中，用下列变量替代，将 $H(z)$转换成 $H(\omega)$。

$$\omega = \frac{\pi}{2}\frac{z-1}{z+1} \tag{6-37}$$

上述方程看起来很熟悉，这是因为它与双线性变换很相近。我们已知双线性变换将 jω 轴转换成单位圆，可以推导出 ω变换同样是将 z 平面的单位圆内部对应于 ω平面的左半部。

在第 4 章中介绍的 $G(s)$的所有设计方法都可以应用在 $G(\omega)$上。在这种处理方法中，首先零阶保持器近似地将连续系统传递函数 $G(s)$转换成离散系统传递函数 $G(z)$，然后用下列变换，从而得到 $G(\omega)$。

$$z = \frac{1+\omega T/2}{1-\omega T/2} \tag{6-38}$$

现在可以用伯德图或根轨迹方法设计补偿器。最后，将设计的补偿器转换回 z 平面。最后一步是非常必要的；因为是用差分方程来构成数字补偿器的。

下面是将连续系统变换到 ω平面的一个例子。

$$H(s) = \frac{1}{s+1}$$

用 c2dm 命令，采用零阶保持器方法，在 T=0. 5s 的采样周期下，将上式数字化为

$$H(z) = \frac{0.393}{z-0.607}$$

用式(6-38)代入上式得到

$$H(\omega) = \frac{1-0.25\omega}{1+1.023\omega}$$

$H(\omega)$的极点和 $H(s)$的极点非常接近。在下一个例子里，将看到，它是采样周期的函数。 $H(s)$是不存在零点的，而 $H(\omega)$在右

半平面有一个零点，这个零点是由采样过程而引入的，是和采样周期相关的，在所有的 ω 变换中，是共同的。因此，在使用伯德图来决定 ω 平面的传递函数的稳定性时必须注意。

可以用 d2cm 命令将 z 平面的传递函数用双线性变换法转换成 ω 平面上的传递函数。同样可以用 c2dm 命令将 ω 平面的传递函数用双线性变换法转换成 z 平面的传递函数。

【例 6-8】 ω 变换。

对于所给的 $H(s)$，用三种不同的采样周期 T=0.5 、 0.1 、 0.05s ，将其转换到 ω 平面，画出连续系统的伯德图和其它三种不同的 ω 变换后的传递函数的伯德图。其中

$$H(s)=\frac{10}{s+5}$$

首先，用 c2dm 命令，采用零阶保持器的方法将其转换成 z 平面的传递函数，然后，再用 d2cm 命令用双线性变换法，转换到 ω 平面上。 MATLAB 程序 6-8 是完成这种转换的程序清单。

```
% MATLAB PROGRAM 6-8
ng=10;dg=[1 5];ts1=0.5;
[ngz1,dgz1]=c2dm(ng,dg,ts1);
[ngw1,dgw1]=d2cm(ngz1,dgz1,ts1,'t');
w=logspace(-1,2,100);
mw1=bode(ngw1,dgw1,w);
```

从而得到 T=0.5s 时

$$H_1(\omega)=1.69\frac{-\omega+4}{\omega+3.393}$$

同样可得到 T=0.1s 时

$$H_2(\omega)=0.49\frac{-\omega+20}{\omega+4.898}$$

T=0.05s 时

$$H_3(\omega)=0.24\frac{-\omega+40}{\omega+4.974}$$

从结果可以得知，当 T 越小时，ω 平面的极点越接近 s 平面

的极点，而且三个传递函数在低频时同原连续系统一样，有着相同的低频增益 2，ω 变换总是保持原来传递函数 $H(s)$的低频增益不变。

在三个传递函数中，有着较大差别的是右半平面的零点的位置。采样频率越高，零点位置与相关频率越远，因此对设计系统的影响也越小。

图 6-17～图 6-20 分别给出了原系统的伯德图和三种不同采样频率下的数字系统伯德图。

由于采样频率不同，造成ω变换后的伯德图有一些差别，这并不说明ω变换是无效的。ω变换精确地反映了用采样信号驱动一个连续系统的实际影响。

在用ω变换设计补偿器之前，再回头详细检查一下ω和 s 之间的关系，回到对ω的定义，令 $z=\mathrm{e}^{sT}$，有

$$\omega=\frac{2}{T}\frac{\mathrm{e}^{sT}-1}{\mathrm{e}^{sT}+1}$$

式中，ω 和 s 都是复合变量，对于连续系统的频率图，令 $s=\mathrm{j}\omega$，在这种情况下，ω 是虚数，由此令ω=jν,故

$$\mathrm{j}\nu=\frac{2}{T}\frac{\mathrm{e}^{\mathrm{j}\omega T}-1}{\mathrm{e}^{\mathrm{j}\omega T}+1}=\mathrm{j}\frac{2}{T}\tan\frac{\omega T}{2}$$

由系统的动态性能要求，决定增益穿越频率 ω_{gc} 之后，由 ω=jν，则下式决定了ω平面的穿越频率

$$\nu_{gc}=\frac{2}{T}\tan\frac{T\omega_{gc}}{2} \tag{6-39}$$

然后，用标准的伯德图设计方法来设计补偿器。

【例 6-9】 ω变换设计。

对于例 6-7 所给的系统及其指标，用ω变换完成补偿器的设计。

首先，在采样周期 T=0.05s 时，将系统离散化，然后用ω变换从而得到ω平面的模型为

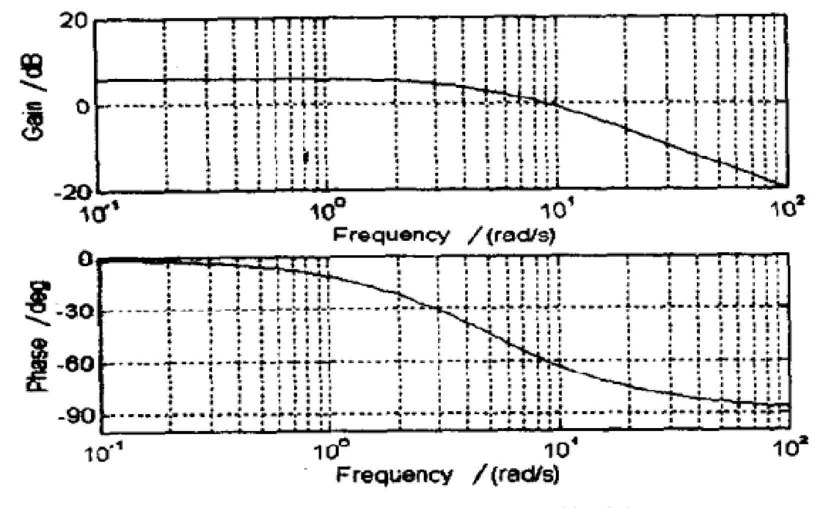


图 6-17 原模拟系统的伯德图

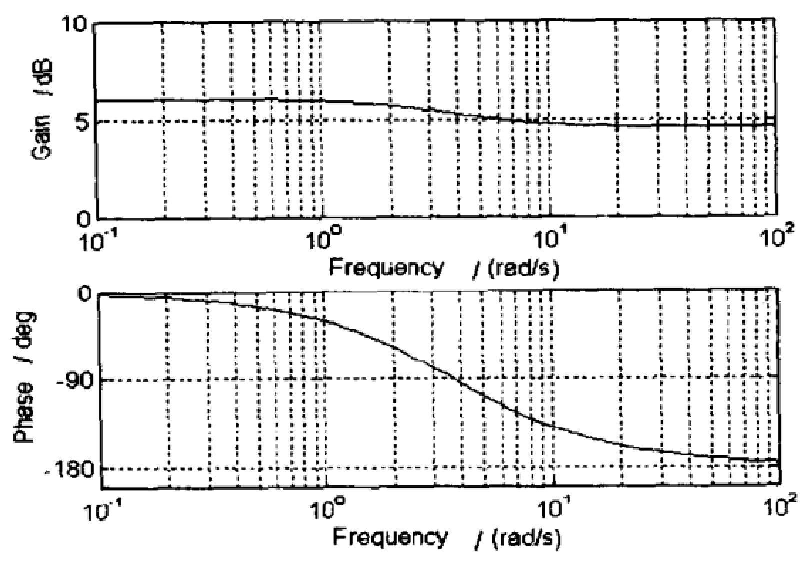


图 6-18 T=0.5s 时数字系统的伯德图

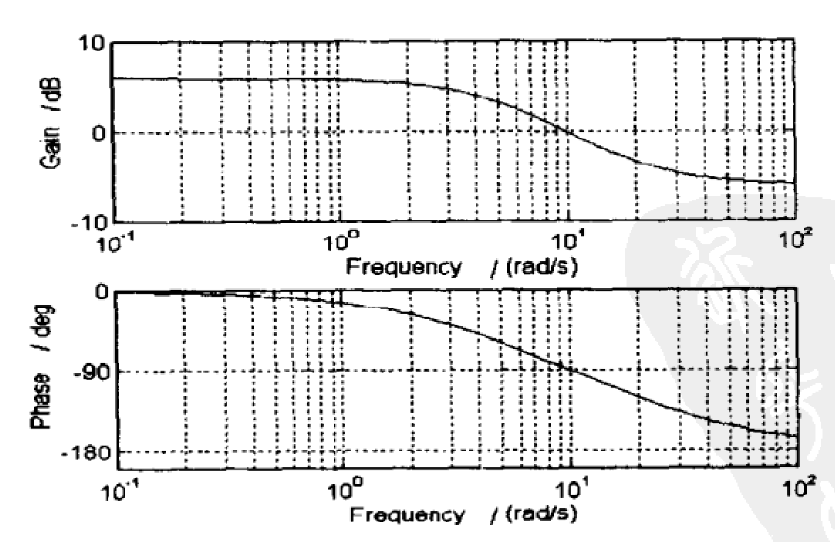


图 6-19 T=0.1s 时数字系统的伯德图

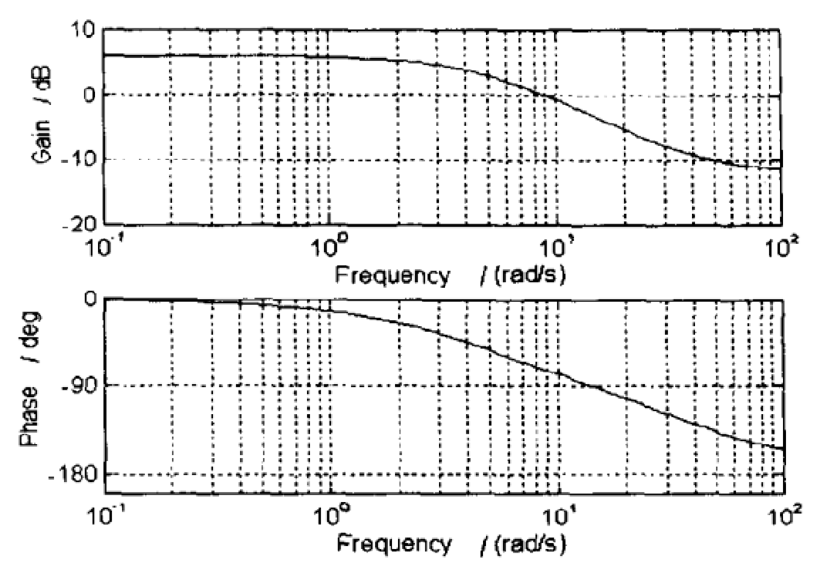


图 6-20 T=0.05s 时数字系统的伯德图

$$G(\omega)=\frac{0.0018\omega^3-0.1261\omega^2-6.8186\omega+362.176}{\omega(\omega^2+28.2814\omega+181.0898)}$$

20 %的稳态误差，要求补偿器的增益为 2.5，要求 ζ=0.5、ω_n=14rad/s，把这些要求转换到频域内，则要求有 50°的相角裕量，幅值穿越频率为 14rad/s，在继续设计之前，将 s 平面的频率转换到ω平面上。

$$\nu_{gc}=\frac{2}{T}\tan\frac{T\omega_{gc}}{2}=14.6$$

用第 4 章介绍的分析伯德图的程序，给出ω平面的超前补偿器如下：

$$K(\omega)=\frac{0.4132\omega+1}{0.0134\omega+1}$$

然后用双线性变换，将其变换回 z 平面，得

$$K(z)=\frac{28.5602z-25.3015}{z+0.3035}$$

该系统的闭环信息有

特征值	幅值	等效阻尼比	等效频率
0.9134	0.9134	1.0000	1.8125
0.4231+0.6425i	0.7693	0.2565	20.4522

0.4231–0.6425i	0.7693	0.2565	20.4522
–0.2534	0.2534	0.4004	68.5673

闭环零点在–2.6170 、 0.8859 、–0.1806 。

MATLAB 程序 6-9 是完成ω变换和设计的程序清单。

```
% MATLAB PROGRAM '6-9
ng=400;dg=[1 30 200 0];ts=0.05;
[ngz,dgz]=c2dm(ng,dg,ts);
[ngw,dgw]=d2cm(ngz,dgz,ts,'t');
nw=[0.4132,1];dw=[0.0134,1];
[nz1,dz1]=c2dm(nw,dw,ts,'t');
nz=nz1*2.5;dz=dz1;
[ngk,dgk]=series(nz,dz,ngz,dgz);
[nt,dt]=cloop(ngk,dgk,-1);
ddamp(dt,ts);
dstep(nt,dt)
axis([0 50 0 1.4])
end
```

加补偿器的反馈系统的阶跃响应见图 6-21 。补偿后系统的超调量为 10 %，调节时间为 2s，复合极点不是主导极点。

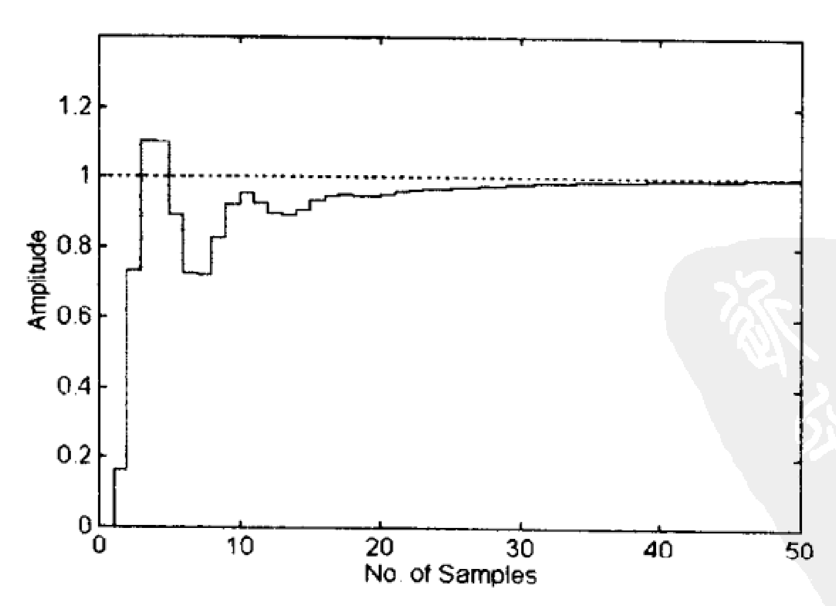


图 6-21 用ω变换设计的补偿后系统的阶跃响应曲线

6.9.5 补偿器的延迟

我们曾经假设，在补偿器的输出和输入之间没有任何延迟，而实际上，这样的补偿器是不存在的。数字补偿器在计算机内是差分方程，因此总有一些处理计算造成的延迟。系统的延迟会引起不稳定，可以预料延迟会影响系统的动态性能和瞬态响应，因此在设计系统时加入已知的延迟是一种好方法。最简单的方法是在系统模型中加入延迟环节，即用 $1/z^n$ 乘以 $G(z)$，对应于延迟 nT 个采样周期，然后继续设计。

6.10 离散状态空间设计简介

离散系统状态空间模型如下：

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k$$
$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{D}\boldsymbol{u}_k$$

其极点是 $\boldsymbol{A}$ 的特征值，同连续系统一样，对离散系统可以采用极点配置方法、线性二次调节器方法、代数设计方法来设计补偿器，控制输入为

$$\boldsymbol{u}_k = -\boldsymbol{K}\boldsymbol{x}_k$$

place 命令对离散系统是完全适用的。系统被离散化，在单位圆内选择期望的极点；被设计的观测器同连续系统是类似的，所不同的是观测器方程是差分方程，而不是微分方程。

其详细的步骤及方法在下一章中介绍。

习　题

6-1 解下列差分方程：

(1) $y_{n+1} + y_n = -\cos(n\pi) \quad (y_0 = 0)$

(2) $y_{n+1} - y_n = x_{n+1} - x_n$ 　　(x_n 是单位阶跃信号)

(3) $y_{n+2} + y_n = 0 \quad y(-1) = 0,\ y(-2) = -2$

6-2 考虑如下的有限脉冲响应滤波器：

$$y_{k+3} = \frac{1}{4}\sum_{i=0}^{3} u_{i+1}$$

(1) 求取其状态空间表达式；

(2) 当 $k>N$ 时，上述滤波器的脉冲响应为 0，求取 N。

6-3 对于下述系统：

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k$$

式中

$$A=\begin{bmatrix}3 & -4\\2 & -3\end{bmatrix} \quad B=\begin{bmatrix}1\\0\end{bmatrix} \quad C=\begin{bmatrix}0 & 1\end{bmatrix}$$

(1) 求其传递函数 $H(z)$；

(2) 求其脉冲响应，绘制其响应曲线；

(3) 假设初始条件为 0，若 $u_k=\delta_k$，求 x_k；

(4) 假设初始条件为 0，求取系统的阶跃响应，并绘制阶跃响应曲线。

6-4 对下列系统：

$$T(s)=\frac{2s^2+0.1s+0.4}{s^2+0.2s+1}$$

采样周期 T=0.1s，用前向差分方法求取状态空间表达式；对上述系统用前向差分、后向差分、双线性变换及零阶保持器方法离散化；求取其阶跃响应和伯德图；当采样周期变为 T=10s 、 T=0.01s 时，用上述方法对系统仿真，求取仿真结果。

6-5 对下列系统：

$$KG(s)=\frac{K}{(s+1)(s+2)}$$

令采样周期 T=1s，用根轨迹和 rlocfind 命令，求取使闭环系统稳定的 K 值。

6-6 对下列系统：

$$G(s)=\frac{1}{s(s+4)}$$

要求对斜坡输入的稳态误差小于 5 %。频带宽度大于 7rad/s，相角裕量大于 45°，用 ω 变换设计补偿器(采样频率大于带宽 10 倍)。

附录

A 程序清单

1. PID 分析设计程序　该程序要求定义如下参数：

$G(s)$=ng/dg， s_1， T=ts， K_i=ki

其 MATLAB 程序清单如下：

```
% MATLAB PROGRAM
%ng;dg;s1;ts;ki
[a,b,c,d]=tf2ss(ng,dg);
[ad,bd]=c2d(a,b,ts);
z1=exp(s1*ts);
kz1=c*inv(z1*eye(size(ad))-ad)*bd+d;
R=-real(1/kz1+(ki*z1*ts)/(z1-1));
X=-imag(1/kz1+(ki*z1*ts)/(z1-1));
alpha=real((z1-1)/(z1*ts));
beta=imag((z1-1)/(z1*ts));
kd=real(X/beta);
kp=real(R-kd*alpha);
nk_pid=[kp*ts+kd+ki*ts*ts -kp*ts-2*kd kd];
dk_pid=[ts -ts 0];
[ngz,dgz]=ss2tf(ad,bd,c,d);
[ngk,dgk]=series(nk_pid,dk_pid,ngz,dgz);
rlocus(ngk,dgk);
axis([-1.1 1.1 -1.1 1.1]);
[nt,dt]=cloop(ngk,dgk);
[mag,wn,zeta]=ddamp(dt,ts)
figure
dstep(nt,dt)
ddamp(dt,ts)
end
```

由以上程序可产生所设计系统的开环根轨迹图和闭环系统的动态响应曲线。

2. 数字超前补偿设计程序　该程序已知如下参数：

$G(s)$=ng/dg， s_1， T=ts， $\overline{K}$ = kbar

其 MATLAB 程序清单如下：

```
% MATLAB PROGRAM
% ng;dg;ts;kbar;s1;
z1=exp(s1*ts);
zbar1=z1-1;
[ngz,dgz]=c2dm(ng,dg,ts);
ngz_z1=polyval(ngz,z1);
dgz_z1=polyval(dgz,z1);
g_z1=ngz_z1/dgz_z1;
w_inv=imag((1+1/(kbar*g_z1))/zbar1)/imag(1/(kbar*g_
z1));
v_inv=-w_inv/(kbar*g_z1)-(1+1/(kbar*g_z1))/zbar1;
w=1/w_inv,v=1/v_inv
a=v-1;
b=w-1;
kc=kbar*w/v;
nzk=[kc,kc*a];
dzk=[1,b];
[ngk,dgk]=series(nzk,dzk,ngz,dgz);
[nt,dt]=cloop(ngk,dgk,-1);
ddamp(dt,ts)
dstep(ngk,dgk)
axis('square'),
zgrid('new'),
rlocus(ngk,dgk)
axis([-1.1 1.1 -1.1 1.1]);
figure
dstep(nt,dt)
end
```

B　z 变换表和 ZOH 表

表 B-1　z 变换表(某些典型信号的 z 变换)

$f(t)$	$f(kT)$	$F(z)$
$\delta(t)$	δ_k	1
$u(t)$	$u(kT)$	$\frac{z}{z-1}$
e^{-at}	e^{-akT}	$\frac{z}{z-e^{-aT}}$
t	kT	$\frac{Tz}{(z-1)^2}$
$\cos\omega t$	$\cos k\omega T$	$\frac{z(z-\cos\omega T)}{z^2-(2\cos\omega T)z+1}$
$\sin\omega t$	$\sin k\omega T$	$\frac{z\sin\omega T}{z^2-(2\cos\omega T)z+1}$
$e^{-at}f(t)$	$e^{-akT}f(kT)$	$F(e^{aT}z)$
$f(t-nT)$	$f((k-n)T)$	$z^{-n}F(z)$
$f(t+nT)$	$f((k+n)T)$	$z^nF(z)-z^nf(0)-\cdots-zf(n-1)$
	$f(\infty)$	$\lim\limits_{z\to1}(z-1)F(z)$

表 B-2　典型传递函数的 z 变换表

$G(s)$	$G(z)$	$G(z)_{ZOH}$
$\frac{1}{s}$	$\frac{z}{z-1}$	$\frac{T}{z-1}$
$\frac{a}{s+a}$	$\frac{az}{z-e^{-aT}}$	$\frac{1-e^{-aT}}{z-e^{-aT}}$
$\frac{a^2}{(s+a)^2}$	$\frac{a^2Tze^{-aT}}{(z-e^{-aT})^2}$	$\frac{Az+B}{(z-e^{-aT})^2}$ $A=1-e^{-aT}-aTe^{-aT}$ $B=(aT-1)e^{-aT}+e^{-2aT}$
$\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$	$\frac{Kze^{-aT}\sin bT}{z^2-2e^{-aT}(\cos bT)z+e^{-2aT}}$ $a=2\zeta\omega_n$ $b=\omega_n\sqrt{1-\zeta^2}$ $K=\omega_n^2/b$	$\frac{Az+B}{z^2-2e^{-aT}(\cos bT)z+e^{-2aT}}$ $A=1-e^{-aT}\cos bT-(a/b)e^{-aT}\sin bT$ $B=e^{-2aT}+(a/b)e^{-aT}-e^{-aT}\cos bT$

第7章　离散系统极点配置和观测器设计方法

7.1　概述

本章将描述离散控制系统极点配置方法和状态观测器的设计。离散系统的极点配置和状态观测器设计的方法与连续系统的设计方法基本相同；决定状态反馈增益矩阵$\boldsymbol{K}$和观测器增益矩阵的方程也完全相同。

从连续系统状态空间方程求取离散系统状态空间方程可以用如下命令：

[G,H]=c2d(A,B,T)

式中，T是离散控制系统的采样周期T，单位是秒。

如果要获取高精度的矩阵$\boldsymbol{G}$和$\boldsymbol{H}$，程序中可以用长字形式命令format long；假如仅仅要求取小数点后四位，可以采用短字形式命令format short。假如程序中不作指定的话，MATLAB产生的矩阵$\boldsymbol{G}$、$\boldsymbol{H}$将是短字形式。

【例7-1】 连续控制系统为

$$\begin{bmatrix}\dot{x}_1\\\dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\-25 & -4\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u \tag{7-1}$$

采样周期T=0.05s，采用如下程序可以求取矩阵$\boldsymbol{G}$、$\boldsymbol{H}$：

```
A=[0 1;-25 -4];
B=[0;1];
[G H]=c2d(A,B,0.05]
```

求得矩阵$\boldsymbol{G}$、$\boldsymbol{H}$为

$$\boldsymbol{G}=\begin{bmatrix}0.9709 & 0.0448\\-1.1212 & 0.7915\end{bmatrix}\quad \boldsymbol{H}=\begin{bmatrix}0.0012\\0.0448\end{bmatrix}$$

注意：离散系统的状态矩阵***G***和控制矩阵***H***是采样周期*T*的函数。对于方程式(7-1)所给的连续系统，采样周期*T*分别为t_{s1}=0.01s、t_{s2}=0.2s、t_{s3}=1s时，用MATLAB程序7-1可以得到不同的矩阵***G***和***H***。

```
% MATLAB PROGRAM 7-1
A=[0 1;-25 -4];B=[0;1];
ts1=0.01;ts2=0.2;ts3=1;
[G1 H1]=c2d(A,B,ts1)
[G2 H2]=c2d(A,B,ts2)
[G3 H3]=c2d(A,B,ts3)
end
G1 =
     0.9988      0.0098
    -0.2450      0.9596
H1 =
     0.0000
     0.0098
G2 =
     0.6401      0.1161
    -2.9017      0.1758
H2 =
     0.0144
     0.1161
G3 =
    -0.0761     -0.0293
     0.7321      0.0410
H3 =
     0.0430
    -0.0293
```

由此可见，采样周期*T*对离散化的矩阵***G***、***H***有较大的影

响。在设计时，必须合理地选取采样周期，以达到满意的性能指标。

【例7-2】 考虑下面的系统

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$

式中

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 20.601 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0.4905 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0.5 \end{bmatrix}$$

假设采样周期T=0.05s，可以得到离散系统状态方程

$$\boldsymbol{x}(k+1) = \boldsymbol{G}\boldsymbol{x}(k) + \boldsymbol{H}\boldsymbol{u}(k)$$

用MATLAB程序7-2可求取矩阵$\boldsymbol{G}$、$\boldsymbol{H}$。

```
% MATLAB PROGRAM 7-2
A=[0 1 0 0;20.601 0 0 0;0 0 0 1;-0.4905 0 0 0];
B=[0;-1;0;0.51];
ts=0.05;
[G H]=c2d(A,B,ts)
end
G =
     1.0259     0.0504          0          0
     1.0389     1.0259          0          0
    -0.0006     0.0000     1.0000     0.0500
    -0.0247    -0.0006          0     1.0000
H =
    -0.0013
    -0.0504
     0.0006
     0.0255
```

7.2 极点配置方法

在这一节里，将讨论用极点配置方法进行离散时间控制系

统的设计。假设所有的状态变量是可测量的和能用于反馈的。假如系统是状态完全能控的，通过加入适当的状态反馈增益矩阵，将闭环系统的极点放置在期望的任意位置上。

假设期望闭环系统的极点是$p_1=\mu_1$、$p_2=\mu_2$、…、$p_n=\mu_n$。当原系统是状态完全能控系统时，可以选取适当的状态反馈增益矩阵$\boldsymbol{K}$，从而使闭环极点在期望的位置上(必须慎重地选取采样周期T，因为采样周期决定着离散化后的矩阵$\boldsymbol{G}$和$\boldsymbol{H}$)。

考虑如图7-1a所示的开环控制系统，其状态方程为

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k) \tag{7-2}$$

式中，$\boldsymbol{x}(k)$为k次采样时刻的状态矢量；$\boldsymbol{u}(k)$为k次采样时刻的控制信号；$\boldsymbol{G}$为$n\times n$矩阵；$\boldsymbol{H}$为$n\times 1$矩阵。

假设控制信号$\boldsymbol{u}(k)$是没有限幅的，控制信号$\boldsymbol{u}(k)$是

$$\boldsymbol{u}(k)=-\boldsymbol{K}\boldsymbol{x}(k)$$

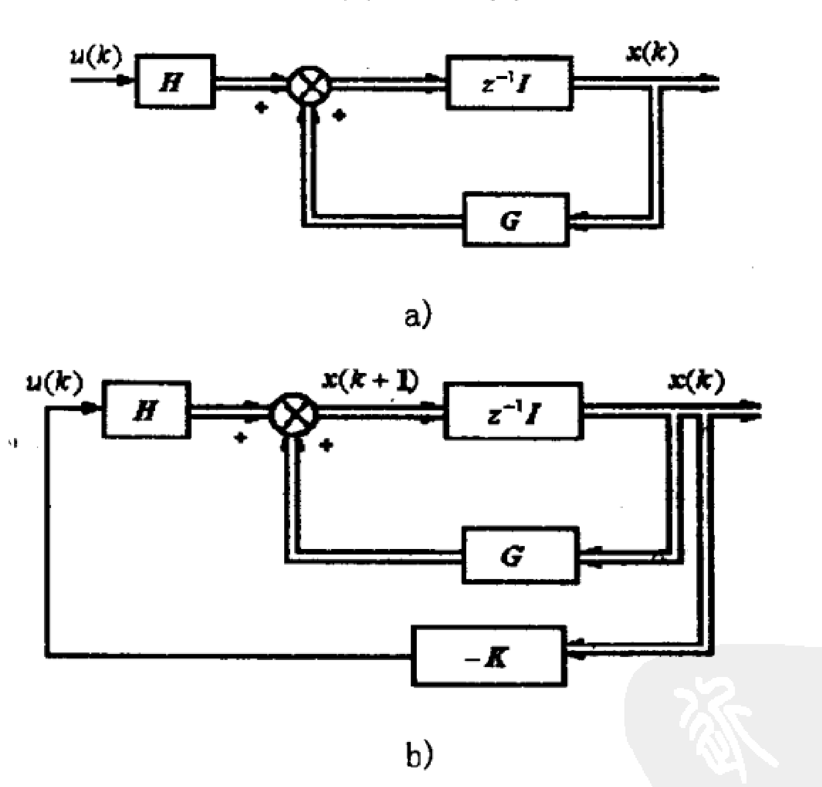


图7-1 控制系统框图

a) 开环 b) 闭环

在上式中，$\boldsymbol{K}$是状态反馈增益矩阵($1\times n$矩阵)，系统变成图7-1b所示的闭环控制系统，其状态方程是

$$x(k+1)=(G-HK)x(k) \tag{7-3}$$

而$G-HK$的特征值是闭环控制系统的期望极点μ_1、μ_2、…、μ_n。极点配置的充分必要条件是，系统是状态完全能控的。

在离散系统中，求取状态反馈增益矩阵的方法与连续系统时是相同的。下面给出求取矩阵K的两种方法：

(1) 状态反馈增益矩阵K为

$$K=\left[\alpha_n-a_n \vdots \alpha_{n-1}-a_{n-1} \vdots \cdots \vdots \alpha_1-a_1\right]T^{-1} \tag{7-4}$$

其中，a_i 是原系统特征方程的系数，其特征方程如下：

$$\left|zI-G\right|=z^n+a_1z^{n-1}+\cdots+a_{n-1}z+a_n=0 \tag{7-5}$$

而α_i 是状态反馈控制系统期望特征方程的系数，其特征方程如下：

$$\left|zI-G+HK\right|=\Phi(z)=z^n+\alpha_1z^{n-1}+\cdots+\alpha_{n-1}z+\alpha_n=0$$

而矩阵T为

$$T=MW \tag{7-6}$$

$$M=\left[H \vdots GH \vdots \cdots \vdots G^{n-1}H\right] \quad W=\begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

假如，系统状态方程已经是可控的标准形式，则状态反馈增益矩阵K是很容易求取的，这是因为转换矩阵T是单位阵，$T=I$，期望的矩阵K是

$$K=\left[\alpha_n-a_n \vdots \alpha_{n-1}-a_{n-1} \vdots \cdots \vdots \alpha_1-a_1\right]$$

(2) 期望的状态反馈增益矩阵K可以由Ackermann公式得到。

$$\begin{aligned} K&=[0\ 0 \cdots 0\ 1]\left[H \vdots GH \vdots \cdots \vdots G^{n-1}H\right]^{-1}\Phi(G) \\ &=[0\ \ 0\ \ \cdots\ \ 0\ \ 1]M^{-1}\Phi(G) \end{aligned} \tag{7-7}$$

式中

$$\Phi(G) = G^n + \alpha_1 G^{n-1} + \cdots + \alpha_{n-1} G + \alpha_n I$$

期望的特征多项式是

$$\Phi(z) = z^n + \alpha_1 z^{n-1} + \cdots + \alpha_{n-1} z + \alpha_n$$

方程式(7-4)和式(7-7)的推导过程与连续系统时是相同的，所不同的是用z代替方程中的s。

7.2.1 说明

状态反馈增益矩阵$\boldsymbol{K}$是基于由干扰引起的误差很快减小到零的原则建立的。矩阵$\boldsymbol{K}$不是唯一的，取决于期望闭环极点的位置，而闭环极点又决定着响应速度。期望闭环极点的位置即特征方程综合反映了误差矢量的响应速度和干扰及噪声的敏感度，即如果提高误差响应速度，则干扰和噪声的影响通常会增加。对于所给的系统，在决定其状态反馈增益矩阵$\boldsymbol{K}$时，选择几个不同的特征方程，建立起相应的矩阵$\boldsymbol{K}$，考查系统的性能，从而选择出最优的方案。

在MATLAB控制工具箱中，有极点配置子程序，可以用命令的形式，直接求取离散系统的状态反馈增益矩阵$\boldsymbol{K}$。该命令格式为

```
K=place(G,H,p)
```

其用法与连续系统极点配置时相同。

采用place命令，可以使离散系统的极点配置更简便。对于求取矩阵$\boldsymbol{K}$的所有步骤，都可以用place命令来取代。

7.2.2 无差拍响应

考虑如下系统：

$$\boldsymbol{x}(k+1) = \boldsymbol{G}\boldsymbol{x}(k) + \boldsymbol{H}\boldsymbol{u}(k)$$

对于状态反馈$\boldsymbol{u}(k) = -\boldsymbol{K}\boldsymbol{x}(k)$，则状态方程变为

$$\boldsymbol{x}(k+1) = (\boldsymbol{G} - \boldsymbol{H}\boldsymbol{K})\boldsymbol{x}(k)$$

上述方程的解为

$$\boldsymbol{x}(k) = (\boldsymbol{G} - \boldsymbol{H}\boldsymbol{K})^k \boldsymbol{x}(0) \tag{7-8}$$

如果矩阵$\boldsymbol{G}-\boldsymbol{H}\boldsymbol{K}$的特征值$\mu_i$在单位圆内，则系统是渐近稳定

的。

假如选择$\boldsymbol{G-HK}$所有的特征值趋向零，可以得到无差拍响应，即

$$\boldsymbol{x}(k)=0 \qquad (k\geqslant q,\ q\leqslant n)$$

在研究无差拍响应时，下面的矩阵是非常重要的。

$$\boldsymbol{N}=\begin{bmatrix}0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & \vdots & & \vdots\\ 0 & 0 & 0 & \cdots & 1\\ 0 & 0 & 0 & \cdots & 0\end{bmatrix}$$

例如，4×4的$\boldsymbol{N}$矩阵为

$$\boldsymbol{N}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}$$

其矩阵的幂分别为

$$\boldsymbol{N}^2=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\quad \boldsymbol{N}^3=\begin{bmatrix}0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\quad \boldsymbol{N}^4=\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}$$

类似地，对于$n\times n$阶矩阵$\boldsymbol{N}$，有

$$\boldsymbol{N}^n=0$$

称矩阵$\boldsymbol{N}$为幂零矩阵。

研究状态完全能控系统

$$\boldsymbol{x}(k+1)=\boldsymbol{Gx}(k)+\boldsymbol{Hu}(k)$$

选择期望的极点位置在原点，即选择期望的特征值为零，$\mu_1=\mu_2=\mu_3=\cdots=\mu_n=0$，则对于任意初始状态$\boldsymbol{x}(0)$，系统的响应是无差拍的。因为期望极点为$\mu_1$、$\mu_2$、…、$\mu_n$的特征多项式为

$$\begin{aligned}|z\boldsymbol{I}-\boldsymbol{G}+\boldsymbol{HK}|&=(z-\mu_1)(z-\mu_2)\cdots(z-\mu_n)\\ &=z^n+\alpha_1 z^{n-1}+\alpha_2 z^{n-2}+\cdots+\alpha_n=z^n\end{aligned}$$

则$\alpha_1=\alpha_2=\cdots=\alpha_n=0$

矩阵$\boldsymbol{K}$被简化为

$$\begin{aligned}\boldsymbol{K} &= [\alpha_n - a_n \vdots \alpha_{n-1} - a_{n-1} \vdots \cdots \vdots \alpha_1 - a_1]\boldsymbol{T}^{-1} \\ &= [-a_n \vdots -a_{n-1} \vdots \cdots \vdots -a_1]\boldsymbol{T}^{-1}\end{aligned} \tag{7-9}$$

由方程式(7-5)所给的转换矩阵$\boldsymbol{T}$，定义

$$\boldsymbol{x}(k) = \boldsymbol{T}\hat{\boldsymbol{x}}(k)$$

$$\boldsymbol{T}^{-1}\boldsymbol{G}\boldsymbol{T} = \hat{\boldsymbol{G}}$$

$$\boldsymbol{T}^{-1}\boldsymbol{H} = \hat{\boldsymbol{H}}$$

由方程式(7-1)所定义的系统可变换为

$$\begin{aligned}\boldsymbol{x}(k+1) &= \boldsymbol{G}\boldsymbol{x}(k) + \boldsymbol{H}\boldsymbol{u}(k) \\ \boldsymbol{T}\hat{\boldsymbol{x}}(k+1) &= \boldsymbol{G}\boldsymbol{T}\hat{\boldsymbol{x}}(k) + \boldsymbol{H}\boldsymbol{u}(k) \\ \hat{\boldsymbol{x}}(k+1) &= \boldsymbol{T}^{-1}\boldsymbol{G}\boldsymbol{T}\hat{\boldsymbol{x}}(k) + \boldsymbol{T}^{-1}\boldsymbol{H}\boldsymbol{u}(k) \\ &= \hat{\boldsymbol{G}}\hat{\boldsymbol{x}}(k) + \hat{\boldsymbol{H}}\boldsymbol{u}(k)\end{aligned}$$

如果用状态反馈$\boldsymbol{u}(k) = -\boldsymbol{K}\boldsymbol{x}(k) = -\boldsymbol{K}\boldsymbol{T}\hat{\boldsymbol{x}}(k)$代入上述方程，则

$$\begin{aligned}\hat{\boldsymbol{x}}(k+1) &= \hat{\boldsymbol{G}}\hat{\boldsymbol{x}}(k) + \hat{\boldsymbol{H}}(-\boldsymbol{K}\boldsymbol{T})\hat{\boldsymbol{x}}(k) \\ &= (\hat{\boldsymbol{G}} - \hat{\boldsymbol{H}}\boldsymbol{K}\boldsymbol{T})\hat{\boldsymbol{x}}(k)\end{aligned} \tag{7-10}$$

参考方程式(7-9)，可得

$$\begin{aligned}\hat{\boldsymbol{G}} - \hat{\boldsymbol{H}}\boldsymbol{K}\boldsymbol{T} &= \hat{\boldsymbol{G}} - \hat{\boldsymbol{H}}[-a_n \ -a_{n-1} \cdots -a_1] \\ &= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}[-a_n \ -a_{n-1} \cdots -a_1] \\ &= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}\end{aligned}$$

因此$\hat{\boldsymbol{G}} - \hat{\boldsymbol{H}}\boldsymbol{K}\boldsymbol{T}$是幂零矩阵，有

$$(\hat{\boldsymbol{G}} - \hat{\boldsymbol{H}}\boldsymbol{K}\boldsymbol{T})^n = 0$$

对原来的状态$\boldsymbol{x}(n)$，可得

$$x(n) = (G - HK)^n x(0) = (T\hat{G}T^{-1} - T\hat{H}K)^n x(0) =$$

$$\left[T(\hat{G} - \hat{H}KT)T^{-1}\right]^n x(0) = T(\hat{G} - \hat{H}KT)^n T^{-1} x(0) = 0$$

因此，假如期望的特征值全部是零，则对任何初始状态$x(0)$，在n个采样周期内，状态矢量变为0，响应是无差拍的。无差拍响应要求控制信号是没有限制的。

7.2.3 无差拍控制的说明

无差拍响应的定义只是针对离散控制系统而言，对连续控制系统没有无差拍的定义。在无差拍控制中，假如控制信号$u(k)$是没有限幅的，在几个采样周期内，非零误差矢量趋于零。因为系统响应至多在几个采样周期时达到稳定，因此过渡过程时间$t_s \leqslant nT$，采样周期T越短，则系统响应的过渡过程时间t_s值越小。这里隐含着控制信号必须有极大的幅值，否则不可能在短时间内使误差响应变为零。

在无差拍控制中，采样周期是唯一的设计参数。因此，假如希望是无差拍控制，设计者首先必须慎重选择采样周期，以使系统在运行时不需要太大的控制信号。实际上，无限制地增大控制信号的幅值是不可能的。假如幅值增大很多的话，就会产生饱和现象，一旦产生饱和现象，则响应就不是无差拍的，过渡过程时间会大于n个采样周期。在实际无差拍控制系统设计中，设计者必须了解控制信号幅值和响应速度之间的利害关系。

【例7-3】 系统状态方程为

$$x(k+1) = Gx(k) + Hu(k)$$

式中

$$G = \begin{bmatrix} 0 & 1 \\ -0.16 & -1 \end{bmatrix} \quad H = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

决定一个适合的状态反馈增益矩阵K，以使系统有如下的闭环极点：$z=0.5 \pm j0.5$

下面首先考察一下能控矩阵的秩

$$[\boldsymbol{H} \vdots \boldsymbol{GH}] = \begin{bmatrix} 0 & 1 \\ 1 & -1 \end{bmatrix}$$

其秩是2。因此系统是状态完全能控的，所以任何的期望极点都是可配置的。期望系统的特征方程是

$$|z\boldsymbol{I} - \boldsymbol{G} + \boldsymbol{HK}| = (z - 0.5 - \mathrm{j}0.5)(z - 0.5 + \mathrm{j}0.5) = z^2 - z + 0.5$$

因此

$$\alpha_1 = -1 \qquad \alpha_2 = +0.5$$

原系统的特征方程是

$$|z\boldsymbol{I} - \boldsymbol{G}| = z^2 + z + 0.16$$

因此

$$a_1 = 1 \qquad a_2 = 0.16$$

下面将用不同的方法解这个问题。

方法一：由方程式(7-3)，状态反馈增益矩阵$\boldsymbol{K}$是

$$\boldsymbol{K} = [\alpha_2 - a_2 \vdots \alpha_1 - a_1]\,\boldsymbol{T}^{-1}$$

$$\boldsymbol{T} = \boldsymbol{MW} = [\boldsymbol{H} \vdots \boldsymbol{GH}] \begin{bmatrix} a_1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}$$

$\boldsymbol{T}$是单位矩阵，而$\boldsymbol{K}$为

$$\boldsymbol{K} = [\alpha_2 - a_2 \vdots \alpha_1 - a_1] = [0.34 \quad -2]$$

MATLAB程序7-3是用方法一计算状态反馈增益矩阵$\boldsymbol{K}$的程序清单。

```
% MATLAB PROGRAM 7-3
G=[0 1;-0.16 -1];H=[0;1];
M=[H G*H];
rank(M)
p=poly(G)
a1=p(2);a2=p(3);
W=[a1 1;1 0];
T=M*W;
J=[0.5+0.5*j 0;0 0.5-0.5*j];
```

```
JJ=poly(J)
aa1=JJ(2);aa2=JJ(3);
K=[aa2-a2 aa1-a1]*(inv(T))
k1=K(1),k2=K(2)
end
```

结果是

```
rank(M) =
        2

p =
    1.0000    1.0000    0.1600
JJ =
    1.0000   -1.0000    0.5000
K =
    0.3400   -2.0000
k1 =
    0.3400
k2 =
    -2
```

方法二：参照Ackermann公式（见方程式(7-7)）有

$$\boldsymbol{K}=\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{H}\vdots\boldsymbol{GH}\end{bmatrix}^{-1}\boldsymbol{\Phi}(\boldsymbol{G})$$

式中

$$\boldsymbol{\Phi}(\boldsymbol{G})=\boldsymbol{G}^2-\boldsymbol{G}+0.5\boldsymbol{I}=\begin{bmatrix}0.34 & -2\\ 0.32 & 2.34\end{bmatrix}$$

因此

$$\boldsymbol{K}=\begin{bmatrix}0.34 & -2\end{bmatrix}$$

MATLAB程序7-4是计算$\boldsymbol{K}$的程序清单：

```
% MATLAB PROGRAM 7-4
G=[0 1;-0.16 -1];H=[0;1];
M=[H G*H];
rank(M)
J=[0.5+0.5*j 0;0 0.5-0.5*j];
```

```
JJ=poly(J)
phi=polyvalm(poly(J),G);
K=[0 1]*(inv(M))*phi
k1=K(1),k2=K(2)
end
rank(M) =
        2
JJ =
    1.0000   -1.0000    0.5000
K =
    0.3400   -2.0000
k1 =
    0.3400
k2 =
    -2
```

方法三：应用下列place命令可以方便地求取状态反馈增益矩阵$\boldsymbol{K}$：

```
G=[0 1;-0.16 -1];
H=[0;1];
p=[0.5+0.5*j 0.5-0.5*j];
K=place(G,H,p)
k1=K(1),k2=K(2)
end
```

【例7-4】对系统

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.16 & -1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \boldsymbol{u}(k)$$

决定状态反馈增益矩阵$\boldsymbol{K}$，以使当控制信号为$\boldsymbol{u}(k)=-\boldsymbol{K}\boldsymbol{x}(k)$时，该系统对初始状态$\boldsymbol{x}(0)$，显示为无差拍响应，假设控制信号$\boldsymbol{u}(k)$是不受限制的。

对于无差拍响应，期望特征方程为

$$|zI-G+HK|=z^2+\alpha_1 z+\alpha_2=z^2$$

式中，$\alpha_1=0$，$\alpha_2=0$。

参考方程式(7-7)，对于无差拍响应，状态反馈增益矩阵K为

$$K=[0\quad 1][G \vdots GH]^{-1}\Phi(G)$$

式中

$$\Phi(G)=G^2$$

MATLAB程序7-5是求取矩阵K的程序清单。

```
% MATLAB PROGRAM 7-5
% Pole placement(Deadbeat response)
%This program determines state feedback gain matrix
G=[0 1;-0.16 -1];
H=[0;1];M=[H G*H];
rank(M)
%Since the rank of M is 2, arbitrary pole placement
%is possible
phi=G^2;
K=[0 1]*inv(M))*phi
k1=K(1)
k2=K(2)
end
```

结果是

```
ans =
     2
K =
    -0.1600   -1.0000
k1 =
    -0.1600
k2 =
    -1
```

7.3 系统设计实例

本节将讨论数字系统调节器的设计、伺服系统控制器的设计。系统框图见图7-2。

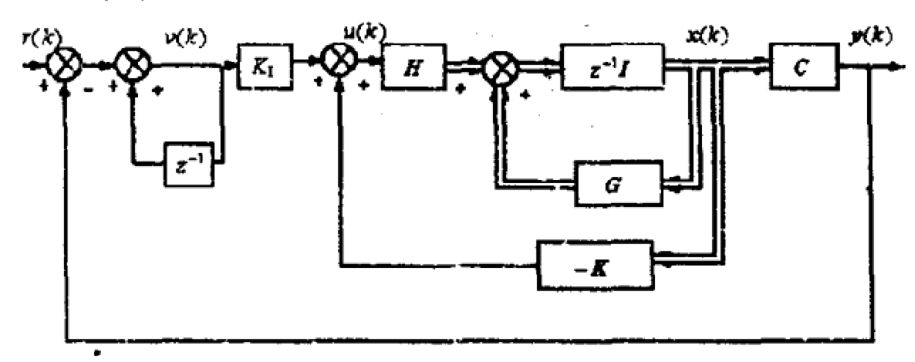


图7-2 系统框图

考虑图7-2所示的伺服系统，原系统并不包含积分器。该系统的设计采用状态反馈和积分控制。

1. 伺服系统设计的基本方法 图7-2所示系统的状态方程和输出方程分别是

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)$$

$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)$$

积分器的方程是

$$\boldsymbol{v}(k)=\boldsymbol{v}(k-1)+\boldsymbol{r}(k)-\boldsymbol{y}(k) \tag{7-11}$$

由于 $\boldsymbol{u}(k)=-\boldsymbol{K}\boldsymbol{x}(k)+K_{\mathrm{I}}\boldsymbol{v}(k)$

则

$$\begin{aligned}&\boldsymbol{v}(k+1)=\boldsymbol{v}(k)+\boldsymbol{r}(k+1)-\boldsymbol{y}(k+1)=\\&\boldsymbol{v}(k)+\boldsymbol{r}(k+1)-\boldsymbol{C}[\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)]=\\&(\boldsymbol{I}-\boldsymbol{C}\boldsymbol{H}K_{\mathrm{I}})\boldsymbol{v}(k)+(-\boldsymbol{C}\boldsymbol{G}+\boldsymbol{C}\boldsymbol{H}\boldsymbol{K})\boldsymbol{x}(k)-\boldsymbol{r}(k+1)\end{aligned}$$

注意：

$$\begin{aligned}&\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}[-\boldsymbol{K}\boldsymbol{x}(k)+K_{\mathrm{I}}\boldsymbol{v}(k)]=\\&(\boldsymbol{G}-\boldsymbol{H}\boldsymbol{K})\boldsymbol{x}(k)+\boldsymbol{H}K_{\mathrm{I}}\boldsymbol{v}(\mathrm{k})\end{aligned}$$

对于整个闭环系统，可以得到下列的状态方程和输出方程：

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G-HK & HK_I \\ -CG+CHK & I-CHK_I \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k+1)$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix}$$

对于阶跃输入$r(k)=r$，则有

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} G-HK & HK_I \\ -CG+CHK & I-CHK_I \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \qquad (7\text{-}12)$$

对于阶跃响应，当k趋向无穷大时，$x(k)$、$u(k)$、$v(k)$趋向常数，由方程式(7-11)得

$$v(\infty) = v(\infty) + r(\infty) - y(\infty)$$

则 $y(\infty) = r(\infty) = r$

没有稳态误差，将$k=\infty$代入方程式(7-12)，可得

$$\begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} G-HK & HK_I \\ -CG+CHK & I-CHK_I \end{bmatrix} \begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix} \qquad (7\text{-}13)$$

定义

$$x_e(k) = x(k) - x(\infty)$$

$$v_e(k) = v(k) - v(\infty)$$

由方程式(7-12)和式(7-13)可推导出

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} G-HK & HK_I \\ -CG+CHK & I-CHK_I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} =$$

$$\begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} \begin{bmatrix} -K & K_I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \qquad (7\text{-}14)$$

定义

$$W(k) = \begin{bmatrix} -K & K_I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} \qquad (7\text{-}15)$$

则方程式(7-14)变为

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} G & 0 \\ -CG & I \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} H \\ -CH \end{bmatrix} W(k) \qquad (7\text{-}16)$$

方程式(7-16)和式(7-15)可以改写为

$$\xi(k+1) = \hat{G}\xi(k) + \hat{H}W(k) \qquad (7\text{-}17)$$

$$W(k) = -K\xi(k) \qquad (7\text{-}18)$$

式中

$$\xi(k)=\begin{bmatrix}x_e(k)\\v_e(k)\end{bmatrix}\quad \hat{G}=\begin{bmatrix}G & 0\\-CG & I\end{bmatrix}$$

$$\hat{H}=\begin{bmatrix}H\\-CH\end{bmatrix}\quad \hat{K}=[K\ \ -K_I]$$

方程式(7-13)和式(7-18)表现为标准的状态空间方程形式。矩阵 $\hat{K}$ 可以用前面介绍的极点配置方法求出。由方程式(7-17)和式(7-18)所表示的系统是状态完全能控的。

【例7-5】倒摆控制系统的设计。

考虑图7-3所示的倒摆系统，倒摆放置在电动机驱动的车子上，在这里仅仅考虑两维问题，即摆只能在纸的平面上运动。倒摆是不稳定的，它随时都可能降下来，除非有一个外力施加在它的上面。假设摆的质量集中在杆的顶端(如图7-3所示)，控制力u施加在车上。

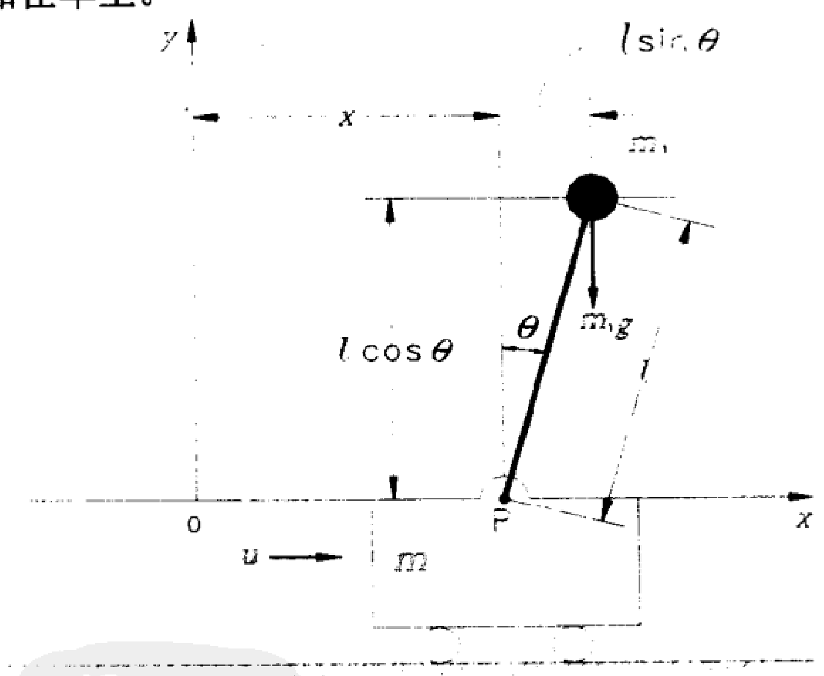


图7-3　倒摆系统

在图中，θ角是杆与垂直线的夹角，假设θ角很小，则可以用θ近似代替$\sin\theta$，$\cos\theta=1$，假设$\dot{\theta}$很小，因此$\theta\dot{\theta}^2=0$。在以上这些假设条件下，系统的非线性方程可以被线性化。

人们希望小车的位置阶跃变化时，保持摆向上。控制力u是

作用于小车的力。为倒摆系统设计数字控制器。

假如将上述系统看作是伺服系统，系统的结构框图将采用图7-2的形式(这意味着设计的控制器包括状态反馈和积分器)。

定义状态变量x_1、x_2、x_3、x_4为

$$x_1 = \theta$$
$$x_2 = \dot{\theta}$$
$$x_3 = x$$
$$x_4 = \dot{x}$$

假设小车的质量m=2kg、摆的质量m_1=0.1kg、杆的长度l=0.5m。

在该系统中，当小车移动时，尽量保持θ角很小。将小车的移动位置作为系统的输出，因此输出方程为

$$y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

倒摆系统的状态方程和输出方程为

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}$$
$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u}$$

式中

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 20.601 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -0.4905 & 0 & 0 & 0 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0.5 \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \qquad \boldsymbol{D} = 0$$

采用状态反馈控制$\boldsymbol{u}=-\boldsymbol{K}\boldsymbol{x}$，设计数字控制器。

因为设计的是数字控制系统，首先将状态方程离散化，用如下的MATLAB命令完成对方程的离散化：

```
[G H]=c2d(A,B,T)
```

式中，T是离散系统的采样周期。在本例中，采样周期T=0.1s，则下列命令：

```
[G H]=c2d(A,B, 0.1)
```

将连续系统的状态空间方程转换为离散系统的状态空间方程。其程序及其结果如下：

```
% MATLAB PROGRAM
A=[0   1   0   0;20.601   0   0   0;
   0   0   0   1;-0.4908   0   0   0];
B=[0;-1; 0;0.5];
[G H]=c2d(A,B,0.1)
end
G =
     1.1048      0.1035           0           0
     2.1316      1.1048           0           0
    -0.0025     -0.0001      1.0000      0.1000
    -0.0508     -0.0025           0      1.0000
H =
    -0.0051
    -0.1035
     0.0025
     0.0501
```

状态空间表达式为

$$
\begin{aligned}
&\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)\\
&\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)+\boldsymbol{D}\boldsymbol{u}(k)\\
&\boldsymbol{v}(k)=\boldsymbol{v}(k-1)+\boldsymbol{v}(k)-\boldsymbol{y}(k)\\
&\boldsymbol{u}(k)=-\boldsymbol{K}\boldsymbol{x}(k)+K_1\boldsymbol{v}(k)
\end{aligned}
$$

整个控制系统的状态方程和输出方程是

$$
\begin{aligned}
&\xi(k+1)=\hat{\boldsymbol{G}}\xi(K)+\hat{\boldsymbol{H}}\boldsymbol{W}(k)\\
&\boldsymbol{W}(k)=-\hat{\boldsymbol{K}}\xi(k)
\end{aligned}
$$

式中

$$
\begin{aligned}
&\xi(k)=\begin{bmatrix}x_1(k) & x_2(k) & x_3(k) & x_4(k) & x_5(k)\end{bmatrix}^{\mathrm{T}}\\
&\hat{\boldsymbol{G}}=\boldsymbol{G}_1=\begin{bmatrix}\boldsymbol{G} & \boldsymbol{0}\\ -\boldsymbol{C}\boldsymbol{G} & \boldsymbol{I}\end{bmatrix}\quad \hat{\boldsymbol{H}}=\boldsymbol{H}_1=\begin{bmatrix}\boldsymbol{H}\\ -\boldsymbol{C}\boldsymbol{H}\end{bmatrix}\\
&\hat{\boldsymbol{K}}=\begin{bmatrix}\boldsymbol{K} & -K_1\end{bmatrix}=\begin{bmatrix}k_1 & k_2 & k_3 & k_4 & -K_1\end{bmatrix}\\
&x_5(k)=\boldsymbol{v}(k)
\end{aligned}
$$

下面的问题是决定矩阵 $\hat{\boldsymbol{K}}$，假设系统的期望闭环极点在

$$\mu_1=0.9+\mathrm{j}0.25\quad \mu_2=0.9-\mathrm{j}0.25\quad \mu_3=\mu_4=\mu_5=0$$

则矩阵G_1和H_1分别是5×5矩阵和5×1矩阵，其MATLAB程序如下：

```
A=[0  1  0  0;20.601  0  0  0;
   0  0  0  1;-0.4908  0  0  0];
B=[0;-1; 0;0.5];
C=[0 0 1 0];
[G H]=c2d(A,B,0.1);
G1=[G  zeros(4,1);-C*G 1]
H1=[H;-C*H]
end
G1 =
     1.1048     0.1035          0          0          0
     2.1316     1.1048          0          0          0
    -0.0025    -0.0001     1.0000     0.1000          0
    -0.0508    -0.0025          0     1.0000          0
     0.0025     0.0001    -1.0000    -0.1000     1.0000
H1 =
    -0.0051
    -0.1035
     0.0025
     0.0501
    -0.0025
```

求取整个系统的可控矩阵的命令为

```
M=[H1 G1*H1 G1^2*H1 G1^3*H1 G1^4*H1]
M =
    -0.0051    -0.0163    -0.0310    -0.0521    -0.0842
    -0.1035    -0.1252    -0.1731    -0.2572    -0.3953
     0.0025     0.0075     0.0126     0.0179     0.0234
     0.0501     0.0506     0.0517     0.0537     0.0570
    -0.0025    -0.0100    -0.0227    -0.0406    -0.0640
```

$\boldsymbol{M}$的秩是5，期望极点的配置是可行的。

MATLAB程序7-7是求取矩阵$\hat{\boldsymbol{K}}$的程序清单。

```
% MATLAB PROGRAM 7-7
A=[0  1  0  0;20.601  0  0  0;0  0  0  1;-0.4908  0
```

```
0  0];
B=[0;-1; 0;0.5];
C=[0 0 1 0];
[G H]=c2d(A,B,0.1);
G1=[G  zeros(4,1);-C*G 1];
H1=[H;-C*H];
M=[H1 G1*H1 G1^2*H1 G1^3*H1 G1^4*H1];
rank(M);
p=poly(G1);
a1=p(2);a2=p(3);a3=p(4);a4=p(5);a5=p(6);
W=[a4 a3 a2 a1 1;a3 a2 a1 1 0;a2 a1 1 0 0;
    a1 1 0 0 0;1 0 0 0 0];
T=M*W;
J=[0.9+0.25*j 0 0 0 0;0 0.9-0.25*j 0 0 0;0 0 0 0 0;
    0 0 0 0 0;0 0 0 0 0];
JJ=poly(J);
aa1=JJ(2);aa2=JJ(3);aa3=JJ(4);aa4=JJ(5);aa5=JJ(6);
KK=[aa5-a5 aa4-a4 aa3-a3 aa2-a2 aa1-a1]*(inv(T))
k1=KK(1),k2=KK(2),k3=KK(3),k4=KK(4),k5=KK(5),KI=-
k5
end
```

状态反馈增益矩阵是

```
KK =
 -371.1863 -103.7552 -236.7408 -168.5171    72.6506
```

由此，已经求得期望系统的状态反馈矩阵和积分器参数，下面分析一下期望系统的阶跃响应。

参照方程式(7-12)可以得到

$$\hat{G}=\begin{bmatrix} G-HK & HK_I \\ -CG+CHK & I-CHK_I \end{bmatrix} \quad \hat{H}=[0\ ;\ 0\ ;\ 0\ ;\ 0\ ;1]$$

$$\hat{C}=[C \quad 0] \qquad \hat{D}=0$$

其方程式为

$$\begin{aligned} x(k+1)&=\hat{G}\,x(k)+\hat{H}\,u(k) \\ y(k)&=\hat{C}\,x(k)+\hat{D}\,u(k) \end{aligned} \tag{7-19}$$

为获得单位阶跃响应$y(k)$，首先将上述方程式(7-19)转换成脉冲传递函数$Y(z)/R(z)$，其MATLAB命令为

$[\text{num den}]=\text{ss2tf}(\hat{G}, \hat{H}, \hat{C}, \hat{D})$

从而得到

```
num=
   0        -0.0000 -0.1818  0.2180   0.2180 -0.1818
den=
   1.0000  -1.8000  0.8725  0.0000  -0.0000  0.0000
```

然后应用命令filter可得输出$\boldsymbol{y}$，其中$\boldsymbol{u}$是单位阶跃响应输入。

y=filter(num,den,u)

如果想得到状态矢量$x_1(k)$的响应曲线，可用下列的输出方程：

$$x_1(k)=\begin{bmatrix}1 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\\x_3(k)\\x_4(k)\\x_5(k)\end{bmatrix}=\boldsymbol{FF}\begin{bmatrix}\boldsymbol{x}(k)\\v(k)\end{bmatrix} \tag{7-20}$$

将方程式(7-19)和式(7-20)的状态空间形式转换成$x_1(z)/R(z)$的脉冲传递函数形式。其命令格式如下：

$[\text{num1 den1}]=\text{ss2tf}(\hat{G}, \hat{H}, \text{FF}, \hat{D})$

然后用filter命令求取x_1的响应曲线，其命令格式如下：

x1=filter(num1,den1,u)

为获得x_2的响应，采用如下的方程形式：

$$x_2(k)=\begin{bmatrix}0 & 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\\x_3(k)\\x_4(k)\\x_5(k)\end{bmatrix}=\boldsymbol{JJ}\begin{bmatrix}\boldsymbol{x}(k)\\v(k)\end{bmatrix} \tag{7-21}$$

将方程式(7-21)和式(7-19)的状态空间形式转换成脉冲传递函数形式，其命令如下：

$[\text{num2 den2}]=\text{ss2tf}(\hat{G}, \hat{H}, \text{JJ}, \hat{D})$

x2=filter(num2,den2,u)

式中，JJ=[0　1　0　0　0]

同理可得

$$[num4\ den4] = ss2tf(\hat{G}, \hat{H}, LL, \hat{D})$$

x4=filter(num4,den4,u)

$$[num5\ den5] = ss2tf(\hat{G}, \hat{H}, MM, \hat{D})$$

x5=filter(num5,den5,u)

式中，LL=[0 0 0 1 0]，MM=[0 0 0 0 1]

MATLAB程序7-8是求取$y(k)$、$x_1(k)$、....、$x_5(k)$的程序清单。

```
% MATLAB PROGRAM 7-8
A=[0  1  0  0;20.601  0  0  0;0  0  0  1;
    -0.4908  0  0  0];
B=[0;-1; 0;0.5];
C=[0 0 1 0];
[G H]=c2d(A,B,0.1);
G1=[G  zeros(4,1);-C*G 1];
H1=[H;-C*H];
M=[H1 G1*H1 G1^2*H1 G1^3*H1 G1^4*H1];
rank(M);
p=poly(G1);
a1=p(2);a2=p(3);a3=p(4);a4=p(5);a5=p(6);
W=[a4 a3 a2 a1 1;a3 a2 a1 1 0;a2 a1 1 0 0;
   a1 1 0 0  0;1 0 0 0 0];
T=M*W;
J=[0.9+0.25*j 0 0 0 0;0 0.9-0.25*j 0 0 0;0 0 0 0 0;
   0 0 0 0 0;0 0 0 0 0];
JJ=poly(J);
aa1=JJ(2);aa2=JJ(3);aa3=JJ(4);aa4=JJ(5);aa5=JJ(6);
KK=[aa5-a5 aa4-a4 aa3-a3 aa2-a2 aa1-a1]*(inv(T));
%k1=KK(1),k2=KK(2),k3=KK(3),k4=KK(4),k5=KK(5),
%Step response of the designed system
K=[KK(1) KK(2) KK(3) KK(4)];
K1=-KK(5);
GG=[G-H*K H*K1;-C*G+C*H*K 1-C*H*K1];
HH=[0;0;0;0;1];CC=[0 0 1 0 0];FF=[1 0 0 0 0];
```

```
JJ=[0 1 0 0 0];LL=[0 0 0 1 0];MM=[0 0 0 0 1];
DD=[0];
[num den]=ss2tf(GG,HH,CC,DD);
r=ones(1,101);
axis([0 100 -0.5 2]);
k=0:100;
y=filter(num,den,r);
plot(k,y,'-'),grid
xlabel('k'),
ylabel('y(k)=x3(k)')
figure
[num1 den1]=ss2tf(GG,HH,FF,DD);
axis([0 100 -1 1]);
x1=filter(num1,den1,r);
plot(k,x1,'-'),grid
xlabel('k'),ylabel('x1(k)')
figure
[num2 den2]=ss2tf(GG,HH,JJ,DD);
axis([0 100 -5 10]);
x2=filter(num2,den2,r);
plot(k,x2,'-'),grid
xlabel('k'),ylabel('x2(k)')
figure
[num4 den4]=ss2tf(GG,HH,LL,DD);
axis([0 100 -5 5]);
x4=filter(num4,den4,r);
plot(k,x4,'-'),grid
xlabel('k'),
ylabel('x4(k)')
figure
[num5 den5]=ss2tf(GG,HH,MM,DD);
axis([0 100 -2 8]);
x5=filter(num5,den5,r);
plot(k,x5,'-'),grid
```

```
xlabel('k')
ylabel('x5(k)=v(k)')
end
```

图7-4是由程序7-8所绘出的小车的位置在单位阶跃输入情况下的响应曲线。

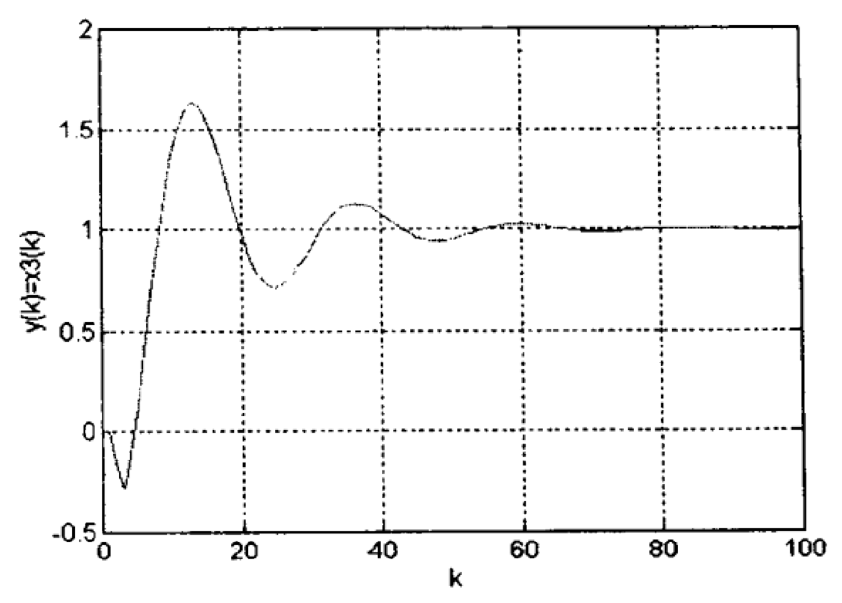


图7-4　小车位置$y(k)$的阶跃响应曲线

从图7-4可以看到，当给小车一个阶跃输入时，小车首先向输入的反方向移动，然后再向输入的方向移动。这种响应是典型的非最小相位系统的响应，从下面的分析中可以认证这一点。

从前面已获得传递函数的num和den可得到

$$\frac{\text{num}}{\text{den}}=\frac{-0.1818z^3+0.2180z^2+0.2180z-0.1818}{z^5-1.8z^4+0.8725z^3}=$$
$$\frac{-0.1818z^{-2}+0.2180z^{-3}+0.2180z^{-4}-0.1818z^{-5}}{1-1.8z^{-1}+0.8725z^{-2}}$$

因为在分子的z^{-2}项的系数是负值，阶跃响应从0到-0. 1818需要三步，如

$$y(0)=0 \qquad y(1)=1 \qquad y(2)=-0.1818$$

输出的稳定值可以由下式获得

$$y(\infty)=\lim_{z\to1}(1-z^{-1})\frac{-0.1818z^{-2}+0.2180z^{-3}+0.2180z^{-4}-0.1818z^{-5}}{(1-1.8z^{-1}+0.8725z^{-2})(1-z^{-1})}=1$$

因此，尽管输出响应向负的方向移动，并停留一段时间，但最终响应回到正的方向，并完全趋向1。这是典型的非最小相位系统的阶跃响应。

图7-5给出了倒摆系统角度的响应曲线。

图7-6给出了倒摆系统角度的响应曲线，图7-7给出了小车的速度响应曲线，图7-8给出了积分器的输出响应曲线。

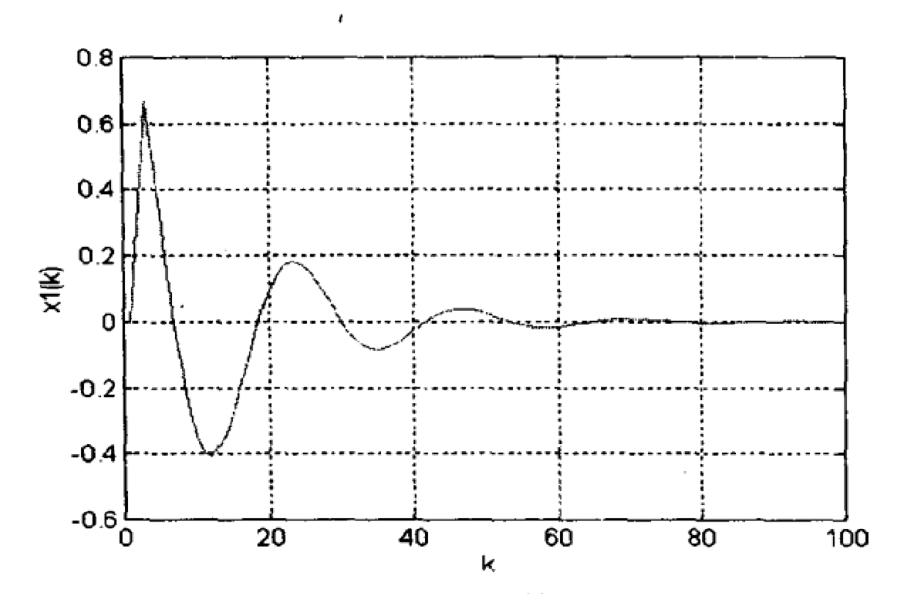


图7-5　状态矢量 $x_1(k)$ 的响应曲线

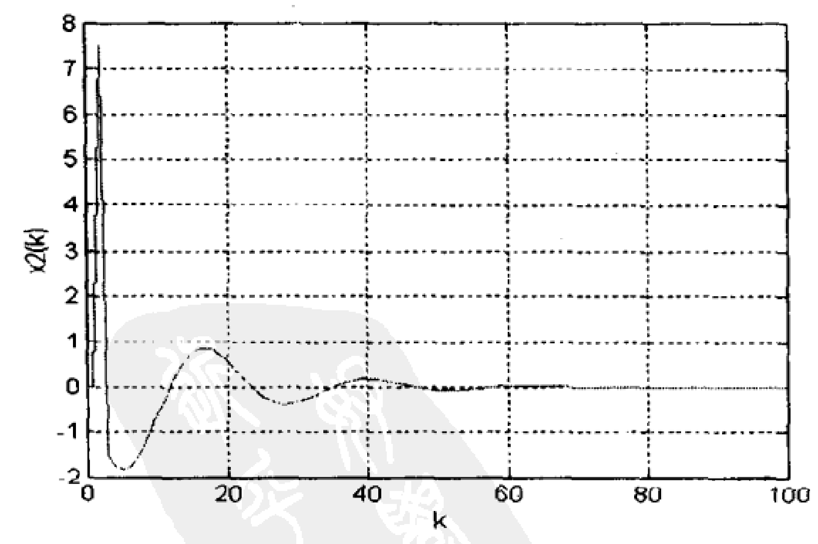


图7-6　角速度 $x_2(k)$ 的响应曲线

由上述响应可以看出，各状态变量和输出都在K>70时达到稳态。采样周期T=0.1s，因此过渡过程时间是KT=7s。

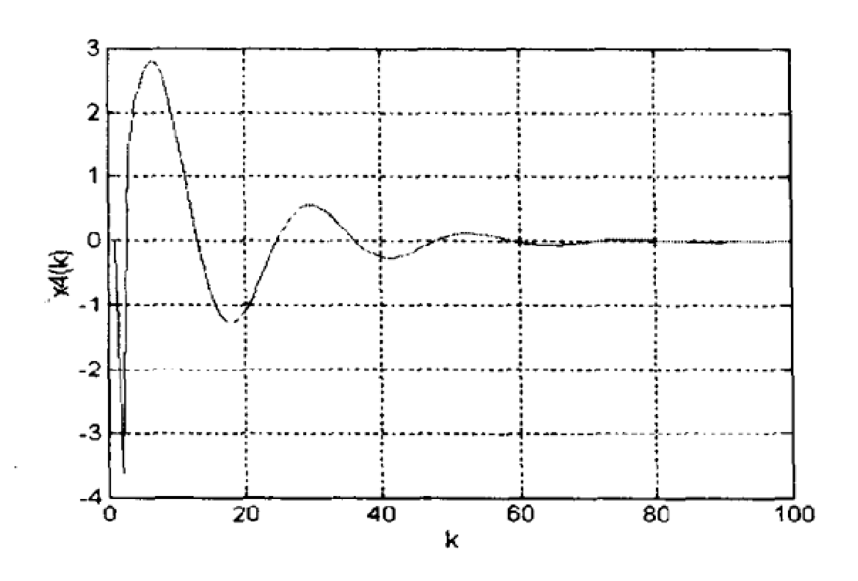


图7-7　小车的速度响应曲线

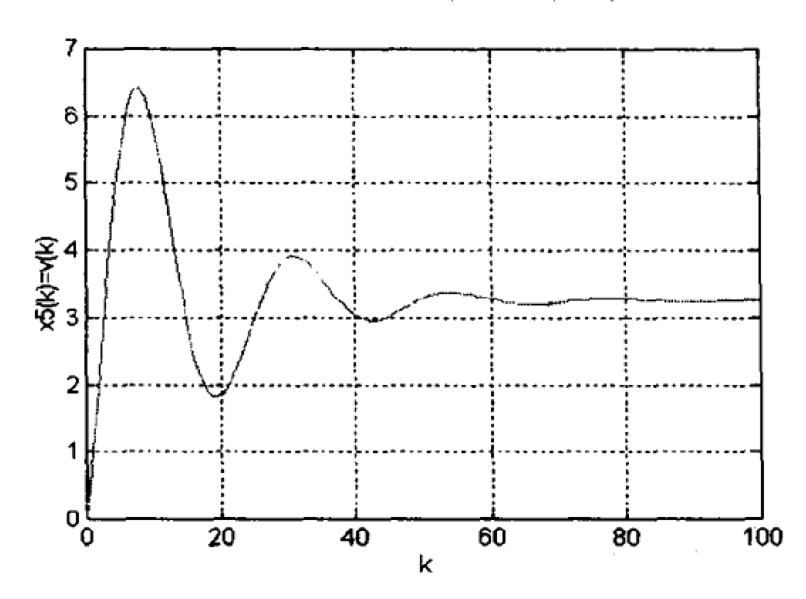


图7-8　积分器的输出响应曲线

7.4　全阶状态观测器的设计

先讨论全阶状态观测器，再讨论最小阶状态观测器。

7.4.1　全阶观测器的结构

状态观测器的阶和系统的阶是相同的，这样的状态观测器叫全阶状态观测器或简称为全阶观测器。

在下面的分析中，假定实际状态$\boldsymbol{x}(k)$是不能被直接测量的。假如要估测状态为$\tilde{\boldsymbol{x}}(k)$，人们希望观测的状态或估算的状态$\tilde{\boldsymbol{x}}(k)$应尽可能接近实际状态$\boldsymbol{x}(k)$。尽管状态观测器没有必要

与原系统有着相同的矩阵$\boldsymbol{G}$和$\boldsymbol{H}$，但是，如果采用相同的矩阵，会简化系统设计。

必须注意：现在的分析中，状态$\boldsymbol{x}(k)$是不能直接测量的，因而估测的状态$\tilde{\boldsymbol{x}}(k)$不能同实际状态$\boldsymbol{x}(k)$比较。而输出$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)$是能够被测量的，可以比较$\tilde{\boldsymbol{y}}(k)=\boldsymbol{C}\tilde{\boldsymbol{x}}(k)$和$\boldsymbol{y}(k)$。若用测量的输出$\boldsymbol{y}(k)$和估测的输出$\tilde{\boldsymbol{y}}(k)$之差用于校正状态$\tilde{\boldsymbol{x}}(k)$，则观测器的动态模型的性能将得到改善。

考虑图7-9所示的状态反馈控制系统，系统的状态方程为

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k) \tag{7-22}$$

$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \tag{7-23}$$

$$\boldsymbol{u}(k)=-\boldsymbol{K}\boldsymbol{x}(k) \tag{7-24}$$

式中，$\boldsymbol{x}(k)$为状态矢量（n维矢量）；$\boldsymbol{u}(k)$为控制矢量（r维矢量）；$\boldsymbol{y}(k)$为输出矢量（m维矢量）。

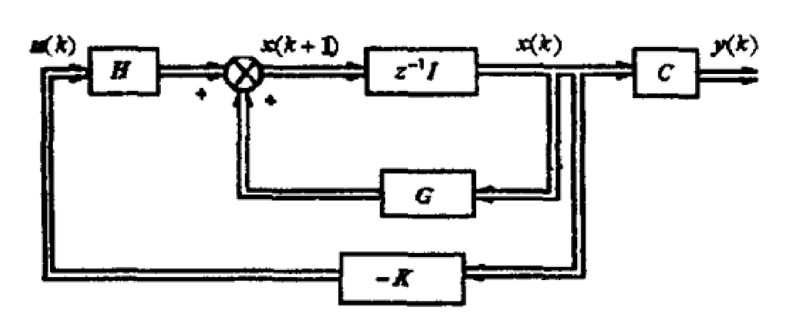


图7-9　状态反馈控制系统

假定系统是完全能观的，而系统状态$\boldsymbol{x}(k)$是不能直接测量的，此时状态观测器系统的结构框图见图7-10。

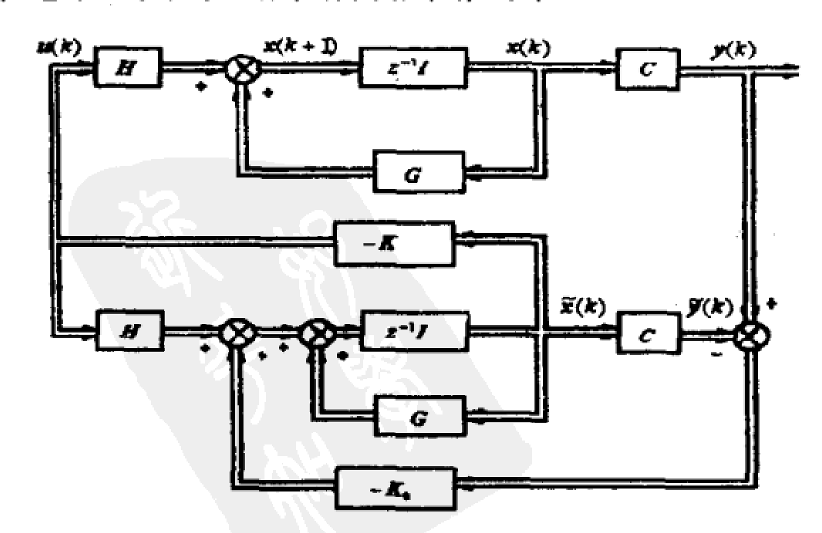


图7-10　状态观测器结构框图

观测器的状态 $\tilde{x}(k)$ 作为控制矢量$u(k)$，即

$$u(k)=-K\tilde{x}(k) \tag{7-25}$$

从图7-10可得

$$\tilde{x}(k+1)=G\tilde{x}(k)+Hu(k)+K_e[y(k)-\tilde{y}(k)] \tag{7-26}$$

式中，K_e是观测器的反馈增益矩阵。上列方程式可改写为

$$\tilde{x}(k+1)=(G-K_eC)\tilde{x}(k)+Hu(k)+K_ey(k) \tag{7-27}$$

由方程式(7-27)所给的状态观测器被称为超前观测器，因为估测的 $\tilde{x}(k+1)$ 是用测量$y(k)$以前一个采样周期的值计算出来的。$G-K_eC$的特征值通常称为观测器极点。

为获得观测器误差方程，用方程式(7-22)减去方程式(7-27)得

$$x(k+1)-\tilde{x}(k+1)=(G-K_eC)[x(k)-\tilde{x}(k)] \tag{7-28}$$

定义$x(k)$与$\tilde{x}(k)$之差为误差$e(k)$，即

$$e(k)=x(k)-\tilde{x}(k)$$

则方程式(7-28)变成

$$e(k+1)=(G-K_eC)e(k) \tag{7-29}$$

从方程式(7-29)可知，误差信号的动态性能是由$G-K_eC$的特征值决定的，假如矩阵$G-K_eC$是稳定矩阵，误差矢量$e(k)$在任何误差的初始条件$e(0)$下，均能趋向零，即估测状态矢量$\tilde{x}(k)$趋向状态矢量$x(k)$。假如矩阵$G-K_eC$的特征值使得误差矢量的动态响应适当地快，则任何误差将很快趋向零。获得快速响应的一种方法是采用无差拍系统，如果所有的特征值都是零，则可以得到无差拍响应。

7.4.2 观测器增益矩阵K_e的方程

同连续系统一样，离散系统的状态观测器增益矩阵K_e可推导为

$$K_e=Q\begin{bmatrix}\alpha_n-a_n\\\alpha_{n-1}-a_{n-1}\\\vdots\\\alpha_1-a_1\end{bmatrix}=(WN')^{-1}\begin{bmatrix}\alpha_n-a_n\\\alpha_{n-1}-a_{n-1}\\\vdots\\\alpha_1-a_1\end{bmatrix} \tag{7-30}$$

式中

$$
\boldsymbol{W}=\begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} \tag{7-31}
$$

$$
\boldsymbol{N}=\left[\boldsymbol{C}' \vdots \boldsymbol{G}'\boldsymbol{C}' \vdots \cdots \vdots (\boldsymbol{G}')^{n-1}\boldsymbol{C}'\right] \tag{7-32}
$$

$$
\boldsymbol{Q}=(\boldsymbol{W}\boldsymbol{N}')^{-1} \tag{7-33}
$$

α_i 是期望特征方程的系数，其特征方程为

$$
z^n+\alpha_1 z^{n-1}+\alpha_2 z^{n-2}+\cdots+\alpha_{n-1}z+\alpha_n=0
$$

a_i 是原系统特征方程的系数，其特征方程为

$$
|z\boldsymbol{I}-\boldsymbol{G}|=z^n+a_1 z^{n-1}+\cdots+a_{n-1}z+a_n=0
$$

假如系统是可观测标准形式，则矩阵($\boldsymbol{WN}'$)成为单位矩阵，即$(\boldsymbol{WN}')^{-1}=\boldsymbol{I}$，矩阵$\boldsymbol{K}_e$就很容易确定。

因为对于无差拍响应，所有的α_i都是零，$\boldsymbol{K}_e$的方程可以简化为

$$
\boldsymbol{K}_e=\begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix}=\boldsymbol{Q}\begin{bmatrix} \delta_n \\ \delta_{n-1} \\ \vdots \\ \delta_1 \end{bmatrix}=(\boldsymbol{W}\boldsymbol{N}')^{-1}\begin{bmatrix} -a_n \\ -a_{n-1} \\ \vdots \\ -a_1 \end{bmatrix} \tag{7-34}
$$

观测器增益矩阵$\boldsymbol{K}_e$的Ackermann公式变为

$$
\boldsymbol{K}_e=\boldsymbol{\Phi}(\boldsymbol{G})\begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{CG} \\ \vdots \\ \boldsymbol{CG}^{n-1} \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{7-35}
$$

式中

$$
\boldsymbol{\Phi}(\boldsymbol{G})=\boldsymbol{G}^n+\alpha_1\boldsymbol{G}^{n-1}+\cdots+\alpha_{n-1}\boldsymbol{G}+\alpha_n\boldsymbol{I}
$$

对于无差拍响应，$\boldsymbol{\Phi}(\boldsymbol{G})$变为

$$
\boldsymbol{\Phi}(\boldsymbol{G})=\boldsymbol{G}^n
$$

【例7-6】考虑系统方程

$$
\begin{aligned} \boldsymbol{x}(k+1)&=\boldsymbol{G}\,\boldsymbol{x}(k)+\boldsymbol{H}u(k) \\ y(k)&=\boldsymbol{C}\boldsymbol{x}(k) \end{aligned}
$$

式中

$$G=\begin{bmatrix}0 & -0.16\\1 & -1\end{bmatrix}\quad H=\begin{bmatrix}0\\1\end{bmatrix}\quad C=\begin{bmatrix}0 & 1\end{bmatrix}$$

设计一个全阶状态观测器，假设系统的结构如图7-10所示，观测器矩阵的期望特征值是

$$z_1=0.5+\mathrm{j}0.5\qquad z_2=0.5+\mathrm{j}0.5$$

因此期望特征方程为

$$(z-0.5+\mathrm{j}0.5)(z-0.5-\mathrm{j}0.5)=z^2-z+0.5=0$$

由于状态观测器设计的结构框图如图7-10所示，因此其主要工作就是确定观测器反馈增益矩阵$\boldsymbol{K}_\mathrm{e}$，首先看一下系统的能观性，矩阵

$$\left[\boldsymbol{C}' \vdots \boldsymbol{G}'\boldsymbol{C}'\right]=\begin{bmatrix}0 & 1\\1 & -1\end{bmatrix}$$

的秩是2，因此系统是完全能控的，可以任意设定观测器的特征方程。

参考方程式(7-29)有

$$\boldsymbol{e}(k+1)=(\boldsymbol{G}-\boldsymbol{K}_\mathrm{e}\boldsymbol{C})\boldsymbol{e}(k)$$
$$\boldsymbol{e}(k)=\boldsymbol{x}(k)-\tilde{\boldsymbol{x}}(k)$$

观测器的特征方程是

$$\left|z\boldsymbol{I}-\boldsymbol{G}+\boldsymbol{K}_\mathrm{e}\boldsymbol{C}\right|=0$$

而$\boldsymbol{K}_\mathrm{e}=\begin{bmatrix}k_1\\k_2\end{bmatrix}$，则特征方程为

$$\left|z\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}-\begin{bmatrix}0 & -0.16\\1 & -1\end{bmatrix}+\begin{bmatrix}k_1\\k_2\end{bmatrix}\begin{bmatrix}0 & 1\end{bmatrix}\right|=0$$

$$z^2+(1+k_2)z+k_1+0.16=0 \tag{7-36}$$

与期望的特征方程比较得

$$k_1=0.34\quad k_2=-2$$

则

$$\boldsymbol{K}_\mathrm{e}=\begin{bmatrix}0.34\\-2\end{bmatrix}$$

MATLAB程序7-9是求取观测器反馈增益矩阵$\boldsymbol{K}_\mathrm{e}$的程序清单。

```
% MATLAB PROGRAM 7-9
```

```
G=[0 -0.16;1 -1];
H=[0;1];
C=[0 1];
N=[C' G'*C'];
rank(N)
J=[0.5+0.5*j  0;0  0.5-0.5*j];
JJ=poly(J)
Phi=polyvalm(poly(J),G);
Ke=Phi*inv(N')*[0;1]
end
```

在离散系统观测器设计中，可以采用控制工具箱中的place命令。从前面的分析可知，离散系统观测器增益矩阵K_e的计算求解过程和连续系统时是一致的，因此可以采用place命令。其格式为

```
Ke=place(G',C',op); Ke=Ke'
```

式中，op为期望极点矢量。

对于上述例子，若采用place命令，MATLAB程序7-9可以简化为MATLAB程序7-10。

```
% MATLAB PROGRAM 7-10
G=[0  -0.16;1  -1];
H=[0;1];
C=[0,1];
op=[0.5+0.5*j  0.5-0.5*j];
Ke=place(G',C',op);
Ke=Ke'
end
```

由上述程序可见，采用MATLAB工具箱中的place命令，可以使观测器增益矩阵K_e的求解大大简化。

7.5 最小阶状态观测器

前面所讨论的状态观测器，要设计重构所有的状态变量。实际上，某些状态变量可能是可以精确测量的，这些可精确测量的状态变量不必被估测。系统的状态变量是n维的，假如观测器的估测状态变量数小于n，则该观测器是降阶观测器。假如降

阶观测器的阶数是尽可能最小的，则该观测器是最小阶状态观测器。

假如状态变量$x(k)$是n维矢量；输出$y(k)$是m维矢量，输出是可测量的。因为m维输出矢量是状态矢量的线性组合，因此m维的状态矢量不需要估测，只需要估测$n-m$维状态矢量。降阶观测器变成$n-m$阶状态观测器，这样的$n-m$阶观测器是最小阶观测器，图7-11给出了最小阶观测器的系统结构框图。

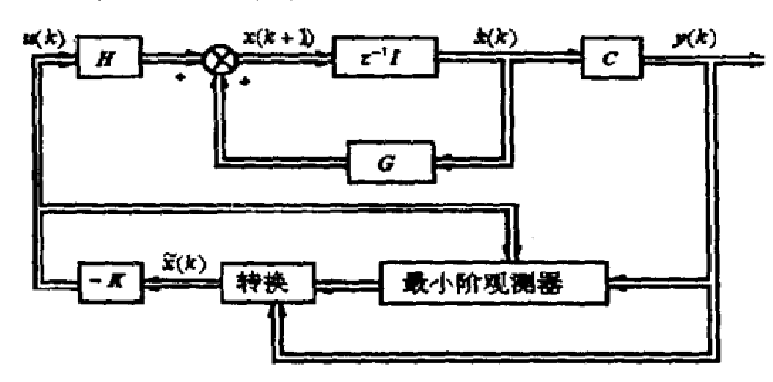


图7-11　最小阶状态观测器结构框图

然而应该注意，假如输出变量的测量含有较大的噪声，而且测量不准确，则采用降阶观测器会产生坏的系统性能。因此，采用全阶观测器可避免输出噪声及测量误差对系统性能的影响。

最小阶状态观测器的设计过程是，将状态矢量$x(k)$分成两部分：

$$x(k)=\begin{bmatrix} x_a(k) \\ \cdots\cdots \\ x_b(k) \end{bmatrix}$$

式中，$x_a(k)$是可以被直接测量的m维状态矢量；$x_b(k)$是不可直接测量的$n-m$维状态矢量。状态方程就可变为如下形式：

$$\begin{bmatrix} x_a(k+1) \\ \cdots\cdots\cdots \\ x_b(k+1) \end{bmatrix}=\begin{bmatrix} G_{aa} & G_{ab} \\ \cdots & \cdots \\ G_{ba} & G_{bb} \end{bmatrix}\begin{bmatrix} x_a(k) \\ \cdots\cdots \\ x_b(k) \end{bmatrix}+\begin{bmatrix} H_a \\ \cdots \\ H_b \end{bmatrix}u(k) \qquad (7\text{-}37)$$

$$y(k)=\begin{bmatrix} I & 0 \end{bmatrix}\begin{bmatrix} x_a(k) \\ \cdots\cdots \\ x_b(k) \end{bmatrix} \qquad (7\text{-}38)$$

式中，G_{aa}为$m\times m$维矩阵；G_{ab}为$m\times(n-m)$维矩阵；G_{ba}为$(n-m)\times m$维矩阵；G_{bb}为$(n-m)\times(n-m)$维矩阵；H_a为$m\times r$维矩阵；H_b

为$(n-m)\times r$维矩阵。

将方程式(7-39)展开重写，状态矢量可测部分的方程变为

$$\boldsymbol{x}_{\mathrm{a}}(k+1)=\boldsymbol{G}_{\mathrm{aa}}\boldsymbol{x}_{\mathrm{a}}(k)+\boldsymbol{G}_{\mathrm{ab}}\boldsymbol{x}_{\mathrm{b}}(k)+\boldsymbol{H}_{\mathrm{a}}\boldsymbol{u}(k)$$

$$\boldsymbol{x}_{\mathrm{a}}(k+1)-\boldsymbol{G}_{\mathrm{aa}}\boldsymbol{x}_{\mathrm{a}}(k)-\boldsymbol{H}_{\mathrm{a}}\boldsymbol{u}(k)=\boldsymbol{G}_{\mathrm{ab}}\boldsymbol{x}_{\mathrm{b}}(k) \qquad (7\text{-}39)$$

方程式(7-39)左边的项是可测的，在设计最小阶状态观测器时，将方程式(7-39)作为输出方程。方程式(7-39)实际上表示了可测状态变量和不可测状态变量之间的关系。

从方程式(7-37)变换可得如下不可测状态的方程：

$$\boldsymbol{x}_{\mathrm{b}}(k+1)=\boldsymbol{G}_{\mathrm{ab}}\boldsymbol{x}_{\mathrm{a}}(k)+\boldsymbol{G}_{\mathrm{bb}}\boldsymbol{x}_{\mathrm{b}}(k)+\boldsymbol{H}_{\mathrm{b}}\boldsymbol{u}(k) \qquad (7\text{-}40)$$

上述方程描述了状态变量中不可测部分的性能。其中$\boldsymbol{G}_{\mathrm{ba}}\boldsymbol{x}_{\mathrm{a}}(k)$和$\boldsymbol{H}_{\mathrm{b}}\boldsymbol{u}(k)$是已知量。

如果采用全阶状态观测器的设计方法，则对最小阶状态观测器的设计就不陌生了。比较全阶观测器和最小阶观测器的状态方程，全阶观测器的状态方程是

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)$$

最小阶状态观测器的状态方程是

$$\boldsymbol{x}_{\mathrm{b}}(k+1)=\boldsymbol{G}_{\mathrm{bb}}\boldsymbol{x}_{\mathrm{b}}(k)+\left[\boldsymbol{G}_{\mathrm{ba}}\boldsymbol{x}_{\mathrm{a}}(k)+\boldsymbol{H}_{\mathrm{b}}\boldsymbol{u}(k)\right]$$

全阶观测器的输出方程是

$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)$$

最小阶状态观测器的输出方程是

$$\boldsymbol{x}_{\mathrm{a}}(k+1)-\boldsymbol{G}_{\mathrm{aa}}\boldsymbol{x}_{\mathrm{a}}(k)-\boldsymbol{H}_{\mathrm{a}}\boldsymbol{u}(k)=\boldsymbol{G}_{\mathrm{ab}}\boldsymbol{x}_{\mathrm{b}}(k)$$

将两组方程比较可得如下的对应关系：

全阶观测器	最小阶观测器
$\boldsymbol{x}(k)$	$\boldsymbol{x}_{\mathrm{b}}(k)$
$\boldsymbol{G}$	$\boldsymbol{G}_{\mathrm{ab}}$
$\boldsymbol{H}\boldsymbol{u}(k)$	$\boldsymbol{G}_{\mathrm{ba}}\boldsymbol{x}_{\mathrm{a}}(k)+\boldsymbol{H}_{\mathrm{b}}\boldsymbol{u}(k)$
$\boldsymbol{y}(k)$	$\boldsymbol{x}_{\mathrm{a}}(k+1)-\boldsymbol{G}_{\mathrm{aa}}\boldsymbol{x}_{\mathrm{a}}(k)-\boldsymbol{H}_{\mathrm{a}}\boldsymbol{u}(k)$
$\boldsymbol{C}$	$\boldsymbol{G}_{\mathrm{ab}}$

因此，最小阶状态观测器的设计参照全阶状态观测器的设计可以有如下的关系。原方程为

$$\tilde{\boldsymbol{x}}(k+1)=(\boldsymbol{G}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{C})\tilde{\boldsymbol{x}}(k)+\boldsymbol{H}\boldsymbol{u}(k)+\boldsymbol{K}_{\mathrm{e}}\boldsymbol{y}(k) \qquad (7\text{-}41)$$

将对应项替换后得

$$\tilde{x}_b(k+1)=(G_{bb}-K_eG_{ab})\tilde{x}_b(k)+G_{ba}x_a(k)+H_bu(k)+ K_e\left[x_a(k+1)-G_{aa}x_a(k)-H_au(k)\right] \tag{7-42}$$

在方程式(7-42)中的观测器反馈增益矩阵K_e是$(n-m)\times m$矩阵，方程式(7-42)决定了最小阶状态观测器的结构。

参考方程式(7-38)，有如下关系：

$$y(k)=x_a(k) \tag{7-43}$$

将方程式(7-43)代入方程式(7-42)有

$$\tilde{x}_b(k+1)=(G_{bb}-K_eG_{ab})\tilde{x}_b(k)+K_ey(k+1)+ (G_{ba}-K_eG_{aa})y(k)+(H_b-K_eH_a)u(k) \tag{7-44}$$

从上式可见，为了估算$\tilde{x}_b(k+1)$，需要测量$y(k+1)$，这不方便，因此希望对上式进行修改。重写方程式(7-44)如下：

$$\begin{aligned}\tilde{x}_b(k+1)-K_ey(k+1)&=(G_{bb}-K_eG_{ab})\tilde{x}_b(k)+\\&\quad(G_{ba}-K_eG_{aa})y(k)+(H_b-K_eH_a)u(k)\\&=(G_{bb}-K_eG_{ab})[\tilde{x}_b(k)-K_ey(k)]+[(G_{bb}-K_eG_{ab})K_e+\\&\quad G_{ba}-K_eG_{aa}]y(k)+(H_b-K_eH_a)u(k)\end{aligned} \tag{7-45}$$

定义

$$x_b(k)-K_ey(k)=x_b(k)-K_ex_a(k)=\eta(k) \tag{7-46}$$

$$\tilde{x}_b(k)-K_ey(k)=\tilde{x}_b(k)-K_ex_a(k)=\tilde{\eta}(k) \tag{7-47}$$

则方程式(7-45)变为

$$\tilde{\eta}(k+1)=(G_{bb}-K_eG_{ab})\tilde{\eta}(k)+[(G_{bb}-K_eG_{ab})K_e+ G_{ba}-K_eG_{aa}]y(k)+(H_b-K_eH_a)u(k) \tag{7-48}$$

方程式(7-47)和式(7-48)决定了最小观测器的动态性能。在上述方程中，为了获得$\tilde{\eta}(k+1)$，并不需要测量$y(k+1)$的值。

下面求取观测器误差方程，定义

$$e(k)=\eta(k)-\tilde{\eta}(k)=x_b(k)-\tilde{x}_b(k) \tag{7-49}$$

由方程式(7-40)减去式(7-42)得

$$x_b(k+1)-\tilde{x}_b(k+1)=G_{bb}\left[x_b(k)-\tilde{x}_b(k)\right]+K_eG_{ab}\tilde{x}_b(k)- K_e\left[x_a(k+1)-G_{aa}x(k)-H_au(k)\right]$$

将方程式(7-39)代入上式得

$$
\begin{aligned}
\boldsymbol{x}_{\mathrm{b}}(k+1)-\tilde{\boldsymbol{x}}_{\mathrm{b}}(k+1) &= \boldsymbol{G}_{\mathrm{bb}}\left[\boldsymbol{x}_{\mathrm{b}}(k)-\tilde{\boldsymbol{x}}_{\mathrm{b}}(k)\right]+ \\
&\quad \boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}}\tilde{\boldsymbol{x}}_{\mathrm{b}}(k)-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}}\boldsymbol{x}_{\mathrm{b}}(k) \\
&= (\boldsymbol{G}_{\mathrm{bb}}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}})\left[\boldsymbol{x}_{\mathrm{b}}(k)-\tilde{\boldsymbol{x}}_{\mathrm{b}}(k)\right]
\end{aligned}
$$

因此上式可变为误差方程

$$
\boldsymbol{e}(k+1)=(\boldsymbol{G}_{\mathrm{bb}}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}})\boldsymbol{e}(k) \tag{7-50}
$$

式中，$\boldsymbol{e}(k)$是$(n-m)$维矢量。误差的性能可以按照期望的特性而设定。参照全阶观测器的设计原理，下列矩阵

$$
\begin{bmatrix} \boldsymbol{G}_{\mathrm{ab}} \\ \boldsymbol{G}_{\mathrm{ab}}\boldsymbol{G}_{\mathrm{bb}} \\ \vdots \\ \boldsymbol{G}_{\mathrm{ab}}\boldsymbol{G}_{\mathrm{bb}}^{\,n-m-1} \end{bmatrix}
$$

的秩是$n-m$，由方程式(7-50)可得最小阶状态观测器的特征方程是

$$
\left|z\boldsymbol{I}-\boldsymbol{G}_{\mathrm{bb}}+\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}}\right|=0 \tag{7-51}
$$

观测器反馈增益矩阵$\boldsymbol{K}_{\mathrm{e}}$可以按下列步骤求取：首先，选择最小阶观测器期望闭环极点的位置；然后，由方程式(7-51)决定$\boldsymbol{K}_{\mathrm{e}}$。其方法与全阶观测器时相似。

例如，假如输出$\boldsymbol{y}(k)$是单输出，则$\boldsymbol{x}_{\mathrm{a}}(k)$是单维的。$\boldsymbol{G}_{\mathrm{ab}}$是$1\times(n-1)$矩阵，$\boldsymbol{G}_{\mathrm{bb}}$是$(n-1)\times(n-1)$矩阵，对于这种情况，Ackermann公式被写成

$$
\boldsymbol{K}_{\mathrm{e}}=\boldsymbol{\Phi}(\boldsymbol{G}_{\mathrm{bb}})\begin{bmatrix} \boldsymbol{G}_{\mathrm{ab}} \\ \boldsymbol{G}_{\mathrm{ab}}\boldsymbol{G}_{\mathrm{bb}} \\ \vdots \\ \boldsymbol{G}_{\mathrm{ab}}\boldsymbol{G}_{\mathrm{bb}}^{\,n-2} \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{7-52}
$$

式中，

$$
\boldsymbol{\Phi}(\boldsymbol{G}_{\mathrm{bb}})=\boldsymbol{G}_{\mathrm{bb}}^{\,n-1}+\alpha_1\boldsymbol{G}_{\mathrm{bb}}^{\,n-2}+\cdots+\alpha_{n-2}\boldsymbol{G}_{\mathrm{bb}}+\alpha_{n-1}\boldsymbol{I} \tag{7-53}
$$

【例7-7】 考虑系统

$$
\begin{aligned}
\boldsymbol{x}(k+1)&=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}u(k) \\
y(k)&=\boldsymbol{C}\boldsymbol{x}(k)
\end{aligned}
$$

式中

$$
\boldsymbol{G}=\begin{bmatrix} 0 & 0 & -0.25 \\ 1 & 0 & 0 \\ 0 & 1 & 0.5 \end{bmatrix} \quad \boldsymbol{H}=\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad \boldsymbol{C}=\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}
$$

假设输出y是可测量的，设计一个最小阶状态观测器，使观测器误差响应是无差拍的。

对本系统 $G_{aa}=0 \quad G_{ab}=\begin{bmatrix}0 & -0.25\end{bmatrix}$

$$G_{ba}=\begin{bmatrix}1\\0\end{bmatrix} \quad G_{bb}=\begin{bmatrix}0 & 0\\1 & 0.5\end{bmatrix}$$

$$H_a=1 \qquad H_b=\begin{bmatrix}0\\1\end{bmatrix}$$

降阶可观矩阵N为

$$N=\left[G_{ab}{}' \vdots G_{bb}{}'G_{ab}{}'\right]=\begin{bmatrix}0 & -0.25\\-0.25 & -0.125\end{bmatrix}$$

矩阵N的秩是2，因此最小阶状态观测器的极点是可以任意配置的。

对无差拍响应，期望的特征方程为

$$\Phi(G_{bb})=G_{bb}{}^2$$

其增益矩阵K_e为

$$K_e=\Phi(G_{bb})\begin{bmatrix}G_{ab}\\G_{ab}G_{bb}\end{bmatrix}^{-1}\begin{bmatrix}0\\1\end{bmatrix}=G_{bb}{}^2(N')^{-1}\begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}0\\-2\end{bmatrix}$$

MATLAB程序7-11是设计最小阶观测器的程序清单。

```
% MATLAB PROGRAM 7-11
% Design of minimum-order observer
Gaa=[0];Gab=[0 -0.25];Gba=[1;0];
Gbb=[0 0;1 0.5];Ha=[1];Hb=[0;1];
N=[Gab' Gbb'*Gab'];
rank(N);
Phi=Gbb^2;
Ke=Phi*inv(N')*[0;1]
end
```

【例7-8】 考虑系统

$$x(k+1)=Gx(k)+Hu(k)$$
$$y(k)=Cx(k)$$

式中

$$G=\begin{bmatrix}0.16 & 2.16\\-0.16 & -1.16\end{bmatrix} \quad H=\begin{bmatrix}-1\\1\end{bmatrix} \quad C=\begin{bmatrix}1 & 1\end{bmatrix}$$

输出$y(k)$可以被测量，决定状态反馈增益矩阵$\boldsymbol{K}$，使系统的闭环极点在

$$z_1=0.6+j0.4 \qquad z_2=0.6-j0.4$$

然后设计最小阶状态观测器。因为$y(k)$是$x_1(k)$和$x_2(k)$的线性组合，我们仅仅需要观测一个状态变量。对于误差性能希望是无差拍响应。

首先考虑系统的能控性和能观性条件。能控性矩阵

$$[\boldsymbol{H} \vdots \boldsymbol{GH}]=\begin{bmatrix}-1 & 2\\ 1 & -1\end{bmatrix}$$

其秩是2。能观性矩阵

$$[\boldsymbol{C}' \vdots \boldsymbol{G}'\boldsymbol{C}']=\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}$$

其秩是2。因此任意的极点配置和观测器的设计是可行的。

为了在本例中应用最小阶状态观测器的设计方法，首先将矩阵$\boldsymbol{C}$的形式转化成[1 0]，因此定义

$$\boldsymbol{x}(k)=\boldsymbol{P}\xi(k)=\begin{bmatrix}1 & -1\\ 0 & 1\end{bmatrix}\xi(k)$$

因此系统方程变为

$$\xi(k+1)=\boldsymbol{P}^{-1}\boldsymbol{GP}\xi(k)+\boldsymbol{P}^{-1}\boldsymbol{H}u(k)$$

$$=\hat{\boldsymbol{G}}\xi(k)+\hat{\boldsymbol{H}}u(k)$$

$$y(k)=\boldsymbol{CP}\xi(k)=\hat{\boldsymbol{C}}\xi(k)$$

式中

$$\hat{\boldsymbol{G}}=\boldsymbol{P}^{-1}\boldsymbol{GP}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}0.16 & 2.16\\ -0.16 & -1.16\end{bmatrix}\begin{bmatrix}1 & -1\\ 0 & 1\end{bmatrix}$$

$$=\begin{bmatrix}0 & 1\\ -0.16 & -1\end{bmatrix}$$

$$\hat{\boldsymbol{H}}=\boldsymbol{P}^{-1}\boldsymbol{H}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}-1\\ 1\end{bmatrix}=\begin{bmatrix}0\\ 1\end{bmatrix}$$

$$\hat{\boldsymbol{C}}=\boldsymbol{CP}=\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1 & -1\\ 0 & 1\end{bmatrix}=\begin{bmatrix}1 & 0\end{bmatrix}$$

因此，在新的状态变量下的系统方程是

$$\begin{bmatrix}\xi_1(k+1)\\ \xi_2(k+1)\end{bmatrix}=\begin{bmatrix}0 & 1\\ -0.16 & -1\end{bmatrix}\begin{bmatrix}\xi_1(k)\\ \xi_2(k)\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}\boldsymbol{u}(k) \tag{7-54}$$

$$\boldsymbol{y}(k)=\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\xi_1(k)\\ \xi_2(k)\end{bmatrix} \tag{7-55}$$

式中，$\xi_1(k)$ 是可测量的状态变量；$\xi_2(k)$ 是不可测量的状态变量，必须被估测。方程式(7-54)和方程式(7-55)与前面介绍的方程式(7-37)和方程式(7-38)是对应的。采用的状态反馈控制方程

$$\boldsymbol{u}(k)=-\boldsymbol{K}\,\boldsymbol{\xi}(k)$$

下面设计状态反馈增益矩阵$\boldsymbol{K}$，因为方程式(7-54)的特征方程是

$$\left|z\boldsymbol{I}-\hat{\boldsymbol{G}}\right|=\begin{vmatrix}z & -1\\ 0.16 & z+1\end{vmatrix}=z^2+z+0.16=0$$

则有$a_1=1$，$a_2=0.16$。

期望的特征方程是

$$\begin{aligned}\left|z\boldsymbol{I}-\hat{\boldsymbol{G}}+\hat{\boldsymbol{H}}\boldsymbol{K}\right|&=(z-0.6-\mathrm{j}\,0.4)(z-0.6+\mathrm{j}\,0.4)\\&=z^2-1.2z+0.52\end{aligned}$$

因此 $\alpha_1=-1.2 \quad \alpha_2=0.52$

因为方程式(7-54)和式(7-55)所定义的系统是能控的标准形，则反馈矩阵$\boldsymbol{K}$为

$$\boldsymbol{K}=\left[\alpha_2-a_2 \vdots \alpha_1-a_1\right]=\begin{bmatrix}0.36 & -2.2\end{bmatrix}$$

下面设计最小阶状态观测器。因为最小阶观测器是一阶的，期望的无差拍响应的特征方程是

$$\Phi(z)=z$$

由方程式(7-54)和方程式(7-55)可得

$$\boldsymbol{G}_{\mathrm{aa}}=0\quad \boldsymbol{G}_{\mathrm{ab}}=1\quad \boldsymbol{G}_{\mathrm{ba}}=-0.16\quad \boldsymbol{G}_{\mathrm{bb}}=-1\quad \boldsymbol{H}_{\mathrm{a}}=0\quad \boldsymbol{H}_{\mathrm{b}}=1$$

参照Ackermann公式，有

$$\boldsymbol{K}_{\mathrm{e}}=\boldsymbol{\Phi}(\boldsymbol{G}_{\mathrm{bb}})\left[\boldsymbol{G}_{\mathrm{ab}}\right]^{-1}[1]=(-1)(1)^{-1}(1)=-1$$

参照方程式(7-49)和式(7-50)，观测器方程为

$$\begin{aligned}&\bar{\xi}_2(k)=\xi_2(k)-\boldsymbol{e}(k)\\&\boldsymbol{e}(k+1)=(\boldsymbol{G}_{\mathrm{bb}}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}})\boldsymbol{e}(k)=(-1+1)\boldsymbol{e}(k)=0\end{aligned}$$

因此，误差至多在一个采样周期内变成零。

下面，求取最小阶观测器的方程。根据$\overline{\eta}(k)$，从方程式(7-48)可得

$$\begin{aligned}\overline{\eta}(k+1) &= (\boldsymbol{G}_{bb}-\boldsymbol{K}_e\boldsymbol{G}_{ab})\overline{\eta}(k)+\left[(\boldsymbol{G}_{bb}-\boldsymbol{K}_e\boldsymbol{G}_{ab})\boldsymbol{K}_e\right.\\ &\quad \left.+\boldsymbol{G}_{ba}-\boldsymbol{K}_e\boldsymbol{G}_{aa}\right]\boldsymbol{y}(k)+(\boldsymbol{H}_b-\boldsymbol{K}_e\boldsymbol{H}_a)\boldsymbol{u}(k)\\ &= -0.16\boldsymbol{y}(k)+\boldsymbol{u}(k)\end{aligned}$$

因此最小阶状态观测器方程是

$$\overline{\eta}(k) = -0.16\boldsymbol{y}(k-1)+\boldsymbol{u}(k-1)$$

被观测的状态反馈控制$\boldsymbol{u}(k)$是

$$\begin{aligned}\boldsymbol{u}(k) &= -\boldsymbol{K}\overline{\xi}(k) = -\begin{bmatrix}0.36 & -2.2\end{bmatrix}\begin{bmatrix}\xi_1(k)\\ \xi_2(k)\end{bmatrix}\\ &= -0.36\xi_1(k)+2.2\xi_2(k)\\ &= -0.36\boldsymbol{y}(k)+2.2\xi_2(k)\end{aligned}$$

由MATLAB程序7-12可计算状态反馈增益矩阵$\boldsymbol{K}$和最小阶观测器的增益矩阵$\boldsymbol{K}_e$。

```
% MATLAB PROGRAM 7-12
% Design of a control system
% Enter matrices G,H,C
G=[0.16 2.16;-0.16 -1.16];
H=[-1;1];C=[1 1];
M=[H G*H];N=[C' G'*C'];
rank(M)
rank(N)
% Hence arbitrary pole placement
% and observer design are possible
P=[1 -1;0 1];
GG=inv(P)*G*P,HH=inv(P)*H,CC=C*P
J=[0.6+j*0.4 0;0 0.6-j*0.4];
poly(J)
Phi=polyvalm(poly(J),GG);
MM=[HH GG*HH];
K=[0 1]*inv(MM)*Phi
```

```
GGaa=0;GGab=1;GGba=-0.16;GGbb=-1;
HHa=0;HHb=1;
Ph=GGbb;
Ke=Ph*inv(GGab)*[1]
end
```

MATLAB程序7-12中，M是系统的能控矩阵，N是系统的能观矩阵。

习　题

7-1　考虑下列系统

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)$$
$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)+\boldsymbol{D}\boldsymbol{u}(k)$$

式中

$$\boldsymbol{G}=\begin{bmatrix}3 & -4\\ 2 & -3\end{bmatrix}\qquad \boldsymbol{H}=\begin{bmatrix}1\\ 0\end{bmatrix}$$

$$\boldsymbol{C}=\begin{bmatrix}1 & 0\end{bmatrix}\qquad \boldsymbol{D}=0$$

若系统的期望闭环极点在[-0.5±j 0.5]上，采用状态反馈，其状态反馈增益矩阵$\boldsymbol{K}$是多少？ 若采用全阶状态观测器， 观测器的期望极点在[-1.5±j 2]，则其观测器增益矩阵$\boldsymbol{K}_e$是多少?

7-2　考虑如下系统

$$G(s)=\frac{2}{s(s+1)}$$

假设采样周期T=0.2，采用状态反馈方法，期望的闭环极点是[-1±j 0.5]，求状态反馈增益矩阵、系统加入反馈后的阶跃响应曲线。求加入观测器后，系统的阶跃响应曲线。

7-3　对于习题7-2，设计一降阶观测器，求取观测器矩阵，并求取系统单位阶跃响应及系统状态及估测状态曲线。



第 8 章　线性二次型最优控制设计方法

8.1　引言

线性二次型最优控制设计方法是 60 年代发展起来的一种应用较多的最优控制系统设计方法。设对象是以状态空间形式给出的线性系统，而目标函数为对象状态和控制输入的二次型函数。二次型问题就是在线性系统的约束条件下，选择控制输入使得二次型目标函数达到最小。

各种问题的二次型公式的优点是它们可以导出易于实现和分析的线性控制率。我们将主要考虑调节器型的问题。假定系统在平衡点处，并期望在有干扰的情况下，也能维持在平衡点（即设定点）。所以，控制目标就是最小化干扰对系统的影响，这也可以与跟踪或伺服机构问题相对比，它们的目标是跟踪给定参考输入或其它外部输入。跟踪问题也可以转换为调节器型问题。

本章首先讨论连续系统的二次型最优控制问题。接着讨论离散系统的二次型最优控制问题，包括有限阶和无限阶（稳态）问题，然后讨论最小能量控制问题。最优观测器设计，即 Kalman-Bucy 滤波器也将在本章讨论。最后，还将讨论二次型高斯问题。

8.2　连续系统的二次型最优控制

设线性定常系统的状态方程为

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{8-1}$$

式中，$x(t)$为 n 维状态矢量；$u(t)$为 r 维控制矢量；A 为 $n \times n$ 维常量矩阵；B 为 $n \times r$ 维常量矩阵，并假定控制矢量 $u(t)$不受

约束。

二次型性能指标为

$$J = \int_{t_0}^{t_f} L(\boldsymbol{x}, \boldsymbol{u}) \mathrm{d}t$$

式中，$L(\boldsymbol{x}, \boldsymbol{u})$为和 $\boldsymbol{x}$ 和 $\boldsymbol{u}$ 的二次型函数。若终端时间 t_f 趋于∞，则系统属于无限长时间状态调节器问题，可以证明，由此可导出线性控制率为

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t) \tag{8-2}$$

式中，$\boldsymbol{K}$ 为 $r \times n$ 维矩阵。

因此，基于这种二次型性能指标的最优控制系统的设计，就简化为矩阵 $\boldsymbol{K}$ 中元素的求取。

具体二次型性能指标如下：

$$J = \int_{0}^{\infty} (\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}'\boldsymbol{R}\boldsymbol{u}) \mathrm{d}\, t \tag{8-3}$$

式中，$\boldsymbol{Q}$ 为 $n \times n$ 维半正定实对称常数矩阵；$\boldsymbol{R}$ 为 $r \times r$ 维正定实对称常数矩阵。最优控制的目标就是求取 $\boldsymbol{u}(t)$，使得性能指标式(8-3)达到最小值。

8.2.1 连续系统二次型调节器问题的求解

求解此类问题有许多种方法。这里采用的是基于 Liapunov 第二方法的求解方法。

通常，先进行控制系统设计，然后再检查其稳定性。但是，也可以先给出稳定性条件，然后再在这些限制条件下设计系统。如果将 Liapunov 第二方法作为最优控制器设计基础的话，则可以保证系统是渐进稳定的。

对于一大类控制系统来说，已经表明在 Liapunov 函数与用于最优控制设计的二次型性能指标之间具有一种直接的关系。

式(8-3)中的矩阵 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 决定了系统误差与控制能量消耗之间的相对重要性。下面来求解此最优控制问题。将式(8-2)代入式(8-1)，可得

$$\dot{x} = Ax - BKx = (A - BK)x$$

在下面的推导中，设矩阵 $A-BK$ 是稳定的，即 $A-BK$ 的特征值具有负实部。

将式(8-2)代入式(8-3)，得

$$J = \int_0^\infty (x'Qx + x'K'RKx)\mathrm{d}t$$

$$= \int_0^\infty x'(Q + K'RK)x\mathrm{d}t$$

设对任意 x 都有

$$x'(Q + K'RK)x = -\frac{\mathrm{d}}{\mathrm{d}t}(x'Px)$$

式中 P 为正定实对称矩阵，则可导出

$$x'(Q + K'RK)x = -\dot{x}'Px - x'P\dot{x}$$

$$= -x'[(A-BK)'P + P(A-BK)]x$$

由 Liapunov 第二方法可知，对于给定的正定矩阵 $Q+K'RK$，如果矩阵 $A-BK$ 是稳定的，则存在正定矩阵 P，使得

$$(A-BK)'P + P(A-BK) = -(Q + K'RK) \tag{8-4}$$

由此可求出 J 为

$$J = \int_0^\infty x'(Q + K'RK)x\mathrm{d}t = -x'Px\Big|_0^\infty$$

$$= -x'(\infty)Px(\infty) + x'(0)Px(0)$$

由于假定 $A-BK$ 的所有特征值都具有负实部，因此有 $x(\infty) \to 0$。这样就得到

$$J = x'(0)Px(0) \tag{8-5}$$

即 J 可由初始条件 $x(0)$ 和 P 得到。

由于 R 为正定实对称矩阵，因此可将 R 写成

$$R = T'T$$

式中 T 为一非奇异矩阵。这样式(8-4)可写成

$$(A' - K'B')P + P(A - BK) + Q + K'T'TK = 0$$

上式经整理还可写成

$$A'+PA+[TK-(T')^{-1}B'P]'[TK-(T')^{-1}B'P]-PBR^{-1}B'P+Q=0$$

J 对 K 取极小值就要求下式对 K 取极小值。

$$x'[TK-(T')^{-1}B'P]'[TK-(T')^{-1}B'P]x$$

由于上式是非负的，其最小值为零，即

$$TK=(T')^{-1}B'P$$

因此可求得最优反馈增益矩阵为

$$K=T^{-1}(T')^{-1}B'P=R^{-1}B'P \tag{8-6}$$

最优控制 $u(t)$ 为

$$u(t)=-Kx(t)=-R^{-1}B'Px(t)$$

式(8-6)中的矩阵 P 必须满足式(8-4)或者满足下列方程：

$$A'P+PA-PBR^{-1}B'P+Q=0 \tag{8-7}$$

式(8-7)称为代数 Riccati(里卡提)方程。

系统的设计步骤如下：

(1)求解式(8-7)Riccati 方程，求得矩阵 P。如果正定矩阵 P 存在(对某些系统，可能不存在正定矩阵 P)，则系统是稳定的或矩阵 $A-BK$ 是稳定的。

(2)将此矩阵 P 代入方程式(8-6)，得到的即为最优反馈增益矩阵 K。

8.2.2 连续系统二次型调节器问题的拓展

二次型调节器问题可以从多方面拓展。这里将讨论其中的两种，即在目标函数中带有交叉乘积项及带有预置稳定度的调节器。

1. **交叉乘积项** 考虑如下更一般化的目标函数

$$J=\int_0^\infty \begin{bmatrix} x \\ u \end{bmatrix}' \begin{bmatrix} Q & N \\ N' & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} dt$$

$$=\int_0^\infty (x'Qx+u'Ru+x'Nu+u'N'x)dt \tag{8-8}$$

当非线性系统进行线性化，或者将一个非线性目标函数用一

个二次型函数近似时，就会出现如上的目标函数形式。还有，若考虑进入系统的功率时，也会用到这种目标函数。另外，在目标函数中包括 $y'y$ 项($y = Cx+Du$)时，也会有如上的 J。

对于这种形式的目标函数，修正后的 Riccati 方程和最优控制器由下式给出：

$$A'P+PA-(PB+N)R^{-1}(PB+N)'+Q = 0 \tag{8-9}$$

$$u = -Kx \quad K = R^{-1}(B'P+N') \tag{8-10}$$

式中，稳定的最优控制存在且唯一的条件为(A, B)完全可控，($A-BR^{-1}N'$, W)完全可观测，而 W 是使 $WW' = Q-NR^{-1}N'$成立的任一矩阵。

2. 带预置稳定度的调节器 若设计一个调节器时，预先给定一个正的标量值α，要求系统的极点位于虚轴左侧，距虚轴为α的地方，则要对目标函数进行修正。修正后的目标函数为

$$J = \int_0^{\infty} e^{2\alpha t}(x'Qx+u'Ru)dt \tag{8-11}$$

对应的 Riccati 方程修正为

$$(A+\alpha I)'P+P(A+\alpha I)+Q-PBR^{-1}B'P = 0 \tag{8-12}$$

8.2.3 MATLAB 实现

命令 lqr 和 lqry 可以直接求解二次型调节器问题，以及相关的 Riccati 方程。这两个命令的格式如下：

```
[ K, P, E ] = lqr(A, B, Q, R, N)
[ K, P, E ] = lqry(A, B, C, D, Q, R)
```

式中，K 为最优反馈增益矩阵；P 为对应 Riccati 方程的唯一正定解(若矩阵 $A-BK$ 是稳定矩阵，则总有 P 的正定解存在)；E 为 $A-BK$ 的特征值。式中的 N 为可选项，其代表交叉乘积项的加权矩阵。lqry 命令用于求解二次型调节器问题的特例，即目标函数中用输出 y 来代替状态 x，即目标函数为

$$J = \int_0^{\infty} (y'Qy+u'Ru)dt \tag{8-13}$$

命令 are 用来求解由下式给出的一般形式的代数 Riccati 方程：

$$A'X+XA-XBX+C=0$$

命令的格式为

X = are(A, B, C)

它返回对应 Riccati 方程的正定解。这个正定解的存在条件为：$\boldsymbol{B}$ 是半正定对称矩阵，$\boldsymbol{C}$ 是对称矩阵。

值得注意的是，有的系统矩阵 $\boldsymbol{A-BK}$ 无论 $\boldsymbol{K}$ 如何选取，也不能稳定。在这种情况下，就不存在 Riccati 方程的正定矩阵 $\boldsymbol{P}$。对于这种情况, lqr 命令就给不出解(参见 MATLAB 程序 8-1)。

【例 8-1】 给定如下系统：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}-1 & 1\\ 0 & 2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}1\\ 0\end{bmatrix}u$$

观察该系统，无论 $\boldsymbol{K}$ 如何选取，也不可能通过反馈控制率

$$u=-\boldsymbol{K}\boldsymbol{x}$$

使系统稳定。

定义 $\boldsymbol{K}=\begin{bmatrix}k_1 & k_2\end{bmatrix}$

则

$$\boldsymbol{A-BK}=\begin{bmatrix}-1 & 1\\ 0 & 2\end{bmatrix}-\begin{bmatrix}1\\ 0\end{bmatrix}\begin{bmatrix}k_1 & k_2\end{bmatrix}=\begin{bmatrix}-1-k_1 & 1-k_2\\ 0 & 2\end{bmatrix}$$

系统特征方程为

$$|s\boldsymbol{I}-\boldsymbol{A}+\boldsymbol{BK}|=\begin{vmatrix}s+1+k_1 & -1+k_2\\ 0 & s-2\end{vmatrix}$$

闭环极点为

$$s_1=-1-k_1 \qquad s_2=2$$

由于 $s_2=2$ 位于 s 右半平面，则无论矩阵 $\boldsymbol{K}$ 如何选取，系统也

是不稳定的。因此该系统不适合用二次型最优控制方法。

假定性能指标中的矩阵 $\boldsymbol{Q}$ 和 $\boldsymbol{R}$ 由下式给出：

$$\boldsymbol{Q}=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \qquad \boldsymbol{R}=[1]$$

由 MATLAB 程序 8-1 得到的解为

$$\boldsymbol{K}=[\ \mathrm{NaN} \quad \mathrm{NaN}\ ]$$

式中， NaN 表示“非数值”。每当二次型最优控制的解不存在时， MATLAB 的矩阵 **K** 就由“ NaN ”组成。

MATLAB 程序 8-1 如下：

```
% MATLAB PROGRAM 8-1
% Design of LQR system
A=[-1 1;0 2];B=[1;0];
Q=[1 0;0 1];R=1;
K=lqr(A,B,Q,R)
Warning: Matrix is singular to working precision.
K =
   NaN   NaN
[K,P,E]=lqr(A,B,Q,R)
Warning: Matrix is singular to working precision.
K =
   NaN   NaN
P =
   -Inf     -Inf
   -Inf     -Inf
E =
   -2.0000
   -1.4142
```

【例 8-2】给定如下系统:

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}$$

式中

$$A=\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\-20 & -11 & -6\end{bmatrix} \qquad B=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

性能指标为

$$J=\int_0^{\infty}(x'Qx+u'Ru)\mathrm{d}t$$

式中

$$Q=\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} \qquad R=[\ 1\ ]$$

试用 MATLAB 来求解 K、P、E。

可用 MATLAB 程序 8-2 来求解 K、P、E：

```
% MATLAB PROGRAM 8-2
% Design of LQR system
A=[0 1 0;0 0 1;-20 -11 -6];B=[0;0;1];
Q=[1 0 0;0 1 0;0 0 1];R=1;
[K,P,E]=lqr(A,B,Q,R)
K =
    0.0250    0.2933    0.1308
P =
    6.1474    2.7689    0.0250
    2.7689    3.2116    0.2933
    0.0250    0.2933    0.1308
E =
  -4.6252
  -0.7528 + 1.9398i
  -0.7528 - 1.9398i
```

【例 8-3】给定如下系统：

$$\dot{x} = Ax+Bu$$

$$y = Cx$$

式中

$$A=\begin{bmatrix}0 & 1 & 0\\0 & 0 & 1\\0 & -2 & -3\end{bmatrix} \qquad B=\begin{bmatrix}0\\0\\1\end{bmatrix} \qquad C=[\ 1 \quad 0 \quad 0\ ]$$

假定系统控制信号由下式给出：

$$u = k_1(r-x_1)-(k_2x_2+k_3x_3)= k_1 r-(k_1x_1+k_2x_2+k_3x_3)$$

其系统结构见图 8-1 。在决定最优控制率时，假定输入信号 $r=0$，试求解反馈增益阵 K

$$K = [\ k_1 \quad k_2 \quad k_3\ ]$$

使得下列性能指标最小化

$$J=\int_0^\infty (x'Qx+u'Ru)\mathrm{d}t$$

式中

$$Q=\begin{bmatrix}q_{11} & 0 & 0\\0 & q_{22} & 0\\0 & 0 & q_{33}\end{bmatrix} \qquad R=1 \qquad x=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix}=\begin{bmatrix}y\\\dot{y}\\\ddot{y}\end{bmatrix}$$

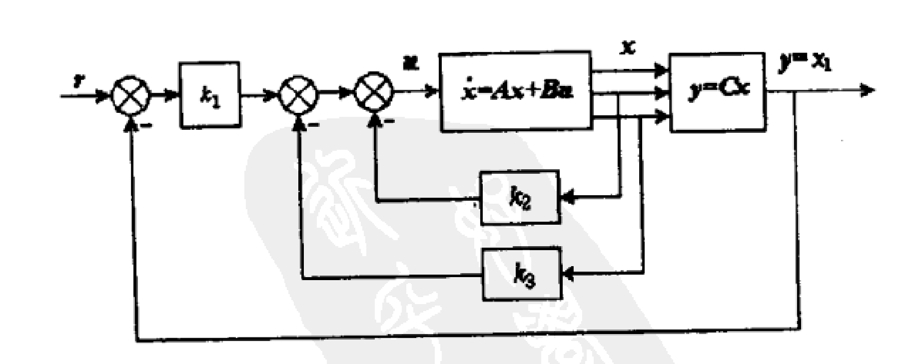


图 8-1　例 8-3 系统结构图

解　为了得到快速响应，q_{11} 与 q_{22}、q_{33} 和 R 相比必须充分

大，因此选择

$$q_{11}=100 \qquad q_{22}=q_{33}=1 \qquad \boldsymbol{R}=[0.01]$$

此例题的 MATLAB 求解过程如 MATLAB 程序 8-3 所示。

```
% MATLAB PROGRAM 8-3
% Design of LQR system
A=[0 1 0;0 0 1;0 -2 -3];B=[0;0;1];
Q=[100 0 0;0 1 0;0 0 1];R=0.01;
K=lqr(A,B,Q,R)
K =
  100.0000    53.1200    11.6711
k1=K(1);k2=K(2);k3=K(3);
AA=A-B*K;BB=B*k1;CC=[1 0 0];DD=0;
[y,x,t]=step(AA,BB,CC,DD);
plot(t,y),  grid
title('Unit-Step Response of LQR System')
xlabel('Time/s')
ylabel('Output  y=x1')
plot(t,x),  grid
title('Response Curves x1,x2,x3 versus t')
xlabel('Time/s')
ylabel('x1,x2,x3')
text(1.6,1.3,'x1')
text(0.7,1.3,'x2')
text(0.35,3.2,'x3')
```

下面再来看一下所设计系统的阶跃响应。系统的状态方程为

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{Ax}+\boldsymbol{Bu} = \boldsymbol{Ax}+\boldsymbol{B}(-\boldsymbol{Kx}+k_1\boldsymbol{r}) \\ &=(\boldsymbol{A}-\boldsymbol{BK})\boldsymbol{x}+\boldsymbol{B}k_1\boldsymbol{r}\end{aligned}$$

输出方程为

$$\boldsymbol{y}=\boldsymbol{Cx}=[\ 1 \quad 0 \quad 0\ ]$$

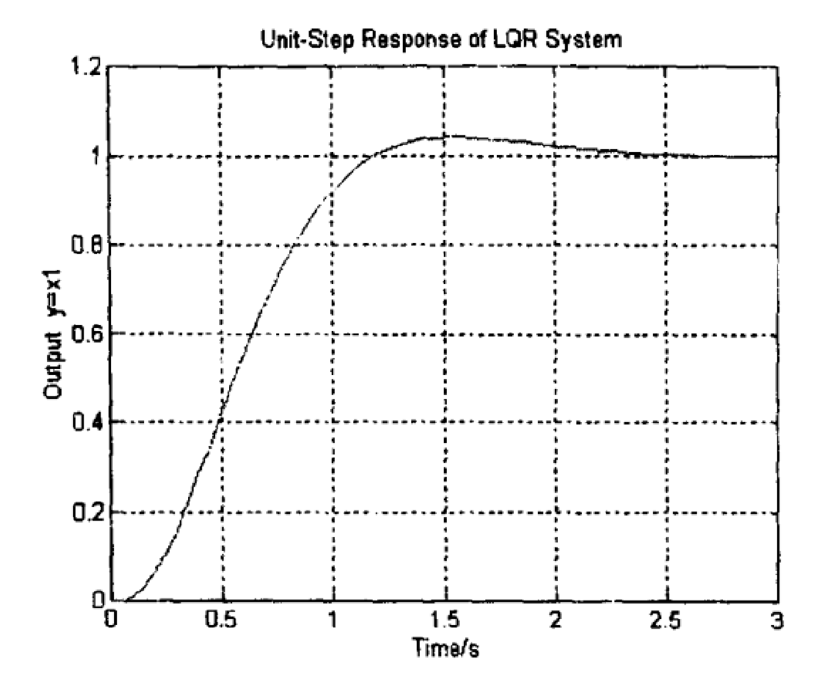


图 8-2　例 8-3 系统输出 y 的阶跃响应曲线

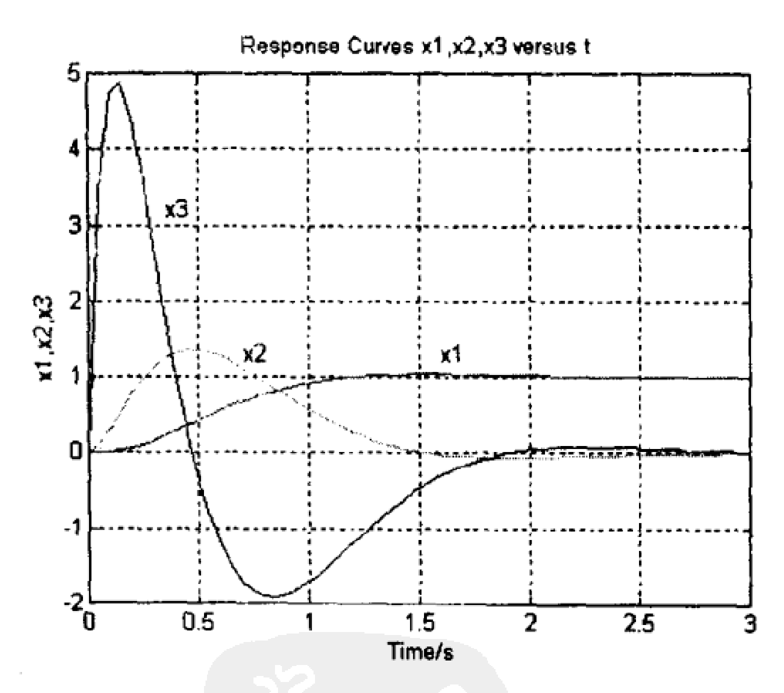


图 8-3　例 8-3 系统状态 x_1、x_2、x_3 的阶跃响应曲线

为了得到阶跃响应，使用如下命令

$$[\ \mathrm{y, x, t}\ ] = \mathrm{step(AA, B\,B, CC, DD)}$$

其中

$$\boldsymbol{AA} = \boldsymbol{A}-\boldsymbol{BK} \qquad \boldsymbol{BB} = \boldsymbol{B}k_1 \qquad \boldsymbol{CC} = \boldsymbol{C} \qquad \boldsymbol{DD} = 0$$

程序 8-3 的后半部分用于获得所设计系统的单位阶跃响应。图 8-2 为输出 y 的阶跃响应曲线，图 8-3 为 x_1、 x_2、 x_3 的阶跃响应曲线。

8.3 离散系统的二次型最优控制

本节首先讨论离散系统的二次型最优控制，并导出有关基本公式，然后再用 MATLAB 程序进行求解。

8.3.1 离散系统二次型最优控制问题的求解

给定线性离散系统

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k) \qquad \boldsymbol{x}(0)=\boldsymbol{x}_0$$
$$(k=0,1,\cdots,N-1) \tag{8-14}$$

式中，$\boldsymbol{x}(k)$为 n 维状态矢量； $\boldsymbol{u}(k)$为 r 维控制矢量； $\boldsymbol{A}$ 为 $n\times n$ 维非奇异矩阵； $\boldsymbol{B}$ 为 $n\times r$ 维矩阵。且假定该系统是状态完全可控的。给出性能指标为

$$J=\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}\sum_{k=0}^{N-1}[\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)] \tag{8-15}$$

式中， $\boldsymbol{Q}$ 为 $n\times n$ 维正定或半正定实对称矩阵； $\boldsymbol{R}$ 为 $r\times r$ 维正定实对称矩阵；$\boldsymbol{S}$ 为 $n\times n$ 维正定或半正定实对称矩阵。$\boldsymbol{Q}$、$\boldsymbol{R}$、$\boldsymbol{S}$ 为加权矩阵，反映了状态矢量 $\boldsymbol{x}(k)$、控制矢量 $\boldsymbol{u}(k)$、终端矢量 $\boldsymbol{x}(N)$的重要性。

离散系统最优控制的要求是，求取最优控制序列$\{\boldsymbol{u}(k)\}$，使得性能指标 J 达到极小值。

系统的初始状态为某个任意状态 $\boldsymbol{x}(0)=\boldsymbol{x}_0$。终端状态 $\boldsymbol{x}(N)$可以是固定的，此时应将式(8-15)中右端第一项$(1/2)\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)$从式(8-15)中去掉。若终端状态 $\boldsymbol{x}(N)$自由，则式(8-15)右端第一项表达了对终端状态的加权，这意味着希望 $\boldsymbol{x}(N)$尽可能地接近原点。

8.3.2 采用离散极小值原理的求解

引入拉格朗日乘子 $\boldsymbol{\lambda}(k)$， 将约束条件即离散状态方程式(8-14)结合到性能指标式(8-15)中，构成了新的性能指标

$$L=\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}\sum_{k=0}^{N-1}\{[\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)]$$

$$+2\,\boldsymbol{\lambda}'(k+1)[\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k)-\boldsymbol{x}(k+1)]\} \tag{8-16}$$

定义哈密尔顿函数为

$$\boldsymbol{H}(k)=\frac{1}{2}[\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)]+\boldsymbol{\lambda}'(k+1)[\boldsymbol{A}\boldsymbol{x}(k)$$

$$+\boldsymbol{B}\boldsymbol{u}(k)-\boldsymbol{x}(k+1)] \tag{8-17}$$

则式(8-16)可改写成

$$L=\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}\sum_{k=0}^{N-1}\{\boldsymbol{H}(k)-\boldsymbol{\lambda}'(k+1)\boldsymbol{x}(k+1)\} \tag{8-18}$$

为使 L 达到极小值，要使 L 对 $\boldsymbol{x}(k)$、 $\boldsymbol{u}(k)$、 $\boldsymbol{\lambda}(k)$求偏导数，并令结果为 0，即

$$\frac{\partial L}{\partial x(k)}=0 \quad (k=1,2,\cdots,N) \tag{8-19}$$

$$\frac{\partial L}{\partial u(k)}=0 \quad (k=0,1,2,\cdots,N-1) \tag{8-20}$$

$$\frac{\partial L}{\partial \lambda(k)}=0 \quad (k=1,2,\cdots,N) \tag{8-21}$$

由此可得到求最优解的必要条件：

$$\frac{\partial L}{\partial \lambda(k)}=0：\ \boldsymbol{A}\boldsymbol{x}(k-1)+\boldsymbol{B}\boldsymbol{u}(k-1)-\boldsymbol{x}(k)=0$$

$$(k=1,2,\cdots,N) \tag{8-22}$$

$$\frac{\partial L}{\partial x(k)}=0：\ \boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{A}'\boldsymbol{\lambda}(k+1)-\boldsymbol{\lambda}(k)=0$$

$$(k=1,2,\cdots,N-1) \tag{8-23}$$

$$\frac{\partial L}{\partial u(k)}=0：\ \boldsymbol{R}\boldsymbol{u}(k)+\boldsymbol{B}'\boldsymbol{\lambda}(k+1)=0$$

$$(k=0,1,\cdots,N-1) \tag{8-24}$$

$$\frac{\partial L}{\partial x(N)}=0: \quad \boldsymbol{Sx}(N)-\boldsymbol{\lambda}(N)=0 \tag{8-25}$$

以上式(8-22)即为离散系统状态方程；式(8-23)又可写成

$$\boldsymbol{\lambda}(k)=\boldsymbol{Qx}(k)+\boldsymbol{A}'\boldsymbol{\lambda}(k+1) \tag{8-26}$$

这一方程称为协态方程。式(8-24)由于 $\boldsymbol{R}^{-1}$ 存在，可写成

$$\boldsymbol{u}(k)=-\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{\lambda}(k+1) \tag{8-27}$$

这是极值条件。式(8-25)可写成

$$\boldsymbol{\lambda}(N)=\boldsymbol{Sx}(N) \tag{8-28}$$

这是横截条件。

将式 (8-27)代入离散系统状态方程式(8-22)或式(8-14)，并考虑到式(8-26)、式(8-28)及初始条件 $\boldsymbol{x}(0)=\boldsymbol{x}_0$，可得到如下两点边值问题：

$$\left.\begin{aligned} \boldsymbol{x}(k+1)&=\boldsymbol{Ax}(k)-\boldsymbol{BR}^{-1}\boldsymbol{B}'\lambda(k+1)\\ \lambda(k)&=\boldsymbol{Qx}(k)+\boldsymbol{A}'\lambda(k+1)\\ \boldsymbol{x}(0)&=\boldsymbol{x}_0\\ \lambda(N)&=\boldsymbol{Sx}(N) \end{aligned}\right\} \tag{8-29}$$

求解式(8-29)的两点边值问题，并将求得的 $\boldsymbol{\lambda}(k+1)$代入式(8-27)，即可得到最优控制序列。这是一种开环控制结构。但是，对于实际应用而言，更希望得到闭环控制结构，以实现性能良好最优反馈控制。

为此，可用数学归纳法来求得 $\boldsymbol{\lambda}(N)$与 $\boldsymbol{x}(N)$之间的关系，即求得 $\boldsymbol{\lambda}(k)$与 $\boldsymbol{x}(k)$之间的关系($k=N$ 时)。设

$$\boldsymbol{\lambda}(k+1)=\boldsymbol{P}(k+1)\,\boldsymbol{x}(k+1) \tag{8-30}$$

将式(8-30)代入式(8-29)，则有

$$\boldsymbol{x}(k+1)=\boldsymbol{Ax}(k)-\boldsymbol{BR}^{-1}\boldsymbol{B}'\boldsymbol{P}(k+1)\boldsymbol{x}(k+1) \tag{8-31}$$

从而得到

$$\boldsymbol{x}(k+1)=[\boldsymbol{I}+\boldsymbol{BR}^{-1}\boldsymbol{B}'\boldsymbol{P}(k+1)]^{-1}\boldsymbol{Ax}(k) \tag{8-32}$$

式中，$\boldsymbol{I}$ 为单位矩阵。将式(8-32)代入式(8-30)，则有

$$\boldsymbol{\lambda}(k+1)=\boldsymbol{P}(k+1)\,[\boldsymbol{I}+\boldsymbol{BR}^{-1}\boldsymbol{B}'\boldsymbol{P}(k+1)]^{-1}\boldsymbol{Ax}(k)$$

$$=[P^{-1}(k+1)+BR^{-1}B']^{-1}Ax(k) \tag{8-33}$$

将式(8-33)代入式(8-26)，可得

$$\lambda(k)=Qx(k)+A'[P^{-1}(k+1)+BR^{-1}B']^{-1}Ax(k)$$

$$=\{Q+A'[P^{-1}(k+1)+BR^{-1}B']^{-1}A\}x(k) \tag{8-34}$$

令

$$P(k)=Q+A'[P^{-1}(k+1)+BR^{-1}B']^{-1}A \tag{8-35}$$

则有

$$\lambda(k)=P(k)x(k) \tag{8-36}$$

由式(8-36)可见，$\lambda(k)$与$x(k)$之间存在线性关系。

将式(8-33)代入式(8-27)，可得

$$u(k)=-R^{-1}B'[P^{-1}(k+1)+BR^{-1}B']^{-1}Ax(k) \tag{8-37}$$

由式(8-35)，有

$$[P^{-1}(k+1)+BR^{-1}B']^{-1}A=(A')^{-1}[P(k)-Q] \tag{8-38}$$

将式(8-38)代入式(8-37)，则有

$$u(k)=-R^{-1}B'(A')^{-1}[P(k)-Q]x(k)$$

$$=-K(k)x(k) \tag{8-39}$$

式中

$$K(k)=R^{-1}B'(A')^{-1}[P(k)-Q] \tag{8-40}$$

式(8-39)为最优控制的闭环形式，它表明$u(k)$与$x(k)$之间存在线性关系。

最优控制矢量$u(k)$可以有几种不同的表达形式。参考式(8-27)和式(8-33)，有

$$u(k)=-R^{-1}B'\lambda(k+1)$$

$$=-R^{-1}B'[P^{-1}(k+1)+BR^{-1}B']^{-1}Ax(k)$$

$$=-K(k)x(k) \tag{8-41}$$

式中

$$K(k)=R^{-1}B'[P^{-1}(k+1)+BR^{-1}B']^{-1}A \tag{8-42}$$

$u(k)$的另一种表达形式为

$$u(k)=-[R+B'P(k+1)B]^{-1}B'P(k+1)Ax(k)$$

$$=-K(k)x(k) \tag{8-43}$$

式中

$$K(k)=[R+B'P(k+1)B]^{-1}B'P(k+1)A \tag{8-44}$$

式(8−44)的推导如下，由于

$$\begin{aligned}&[R+B'P(k+1)B]^{-1}B'P(k+1)[P^{-1}(k+1)+BR^{-1}B']\\&=[R+B'P(k+1)B]^{-1}B'[I+P(k+1)BR^{-1}B']\\&=[R+B'P(k+1)B]^{-1}[R+B'P(k+1)B]R^{-1}B'\\&=R^{-1}B'\end{aligned}$$

因此有

$$\begin{aligned}&R^{-1}B'[P^{-1}(k+1)+BR^{-1}B']^{-1}\\&=[R+B'P(k+1)B]^{-1}B'P(k+1)\end{aligned}$$

由式(8−40)、式(8−42)、式(8−44)给出的 $K(k)$表达式是相等的。而且 $K(k)$具有这样一个性质，即其在临近终端时刻是时变的，在其它情况下，几乎是一个常数。

实际上，式(8−35)是一个 $n\times n$ 维矩阵一阶非线性差分方程，通常称为 Riccati 差分方程。

8.3.3 最小性能指标的计算

$$\min J=\min\left\{\frac{1}{2}x'(N)Sx(N)+\frac{1}{2}\sum_{k=0}^{N-1}[x'(k)Qx(k)+u'(k)Ru(k)]\right\}$$

由式(8−29)和式(8−30)可得

$$P(k)x(k)=Qx(k)+A'P(k+1)x(k+1)$$

上式等号两边左乘 $x'(k)$,有

$$x'(k)P(k)x(k)=x'(k)Qx(k)+x'(k)A'P(k+1)x(k+1)$$

再将式(8−33)代入上式，有

$$\begin{aligned}x'(k)P(k)x(k)&=x'(k)Qx(k)+x'(k+1)[I+BR^{-1}B'P(k+1)]'\times\\&\quad P(k+1)x(k+1)\\&=x'(k)Qx(k)+x'(k+1)[I+P(k+1)BR^{-1}B']'\times\\&\quad P(k+1)x(k+1)\end{aligned}$$

因此

$$\begin{aligned}x'(k)Qx(k)=&\,x'(k)P(k)x(k)-x'(k+1)P(k+1)x(k+1)\\&-x'(k+1)P(k+1)BR^{-1}B'P(k+1)x(k+1)\end{aligned}$$

$$(8\text{-}45)$$

再由式(8-27)，可导出

$$\begin{aligned}\boldsymbol{u}'(k)\boldsymbol{R}\boldsymbol{u}(k)&=[-\boldsymbol{x}'(k+1)\boldsymbol{P}(k+1)\boldsymbol{B}\boldsymbol{R}^{-1}]\boldsymbol{R}[-\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{P}(k+1)\boldsymbol{x}(k+1)]\\&=\boldsymbol{x}'(k+1)\boldsymbol{P}(k+1)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{P}(k+1)\boldsymbol{x}(k+1)\end{aligned}\tag{8-46}$$

将式(8-45)与式(8-46)相加，得

$$\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)=\boldsymbol{x}'(k)\boldsymbol{P}(k)\boldsymbol{x}(k)-\boldsymbol{x}'(k+1)\boldsymbol{P}(k+1)\boldsymbol{x}(k+1)\tag{8-47}$$

在将式(8-47)代入性能指标的式(8-15)，得

$$\begin{aligned}J_{\min}&=\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}\sum_{k=0}^{N-1}[\boldsymbol{x}'(k)\boldsymbol{P}(k)\boldsymbol{x}(k)\\&\quad-\boldsymbol{x}'(k+1)\boldsymbol{P}(k+1)\boldsymbol{x}(k+1)]\\&=\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}[\boldsymbol{x}'(0)\boldsymbol{P}(0)\boldsymbol{x}(0)-\boldsymbol{x}'(1)\boldsymbol{P}(1)\boldsymbol{x}(1)\\&\quad+\boldsymbol{x}'(1)\boldsymbol{P}(1)\boldsymbol{x}(1)-\boldsymbol{x}'(2)\boldsymbol{P}(2)\boldsymbol{x}(2)+\cdots\\&\quad+\boldsymbol{x}'(N-1)\boldsymbol{P}(N-1)\boldsymbol{x}(N-1)-\boldsymbol{x}'(N)\boldsymbol{P}(N)\boldsymbol{x}(N)]\\&=\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{S}\boldsymbol{x}(N)+\frac{1}{2}\boldsymbol{x}'(0)\boldsymbol{P}(0)\boldsymbol{x}(0)-\frac{1}{2}\boldsymbol{x}'(N)\boldsymbol{P}(N)\boldsymbol{x}(N)\end{aligned}\tag{8-48}$$

再由式(8-28)和式(8-36)，有 $\boldsymbol{P}(N)=\boldsymbol{S}$ 。因此，上式就变成

$$J_{\min}=\frac{1}{2}\boldsymbol{x}'(0)\boldsymbol{P}(0)\boldsymbol{x}(0)\tag{8-49}$$

【例 8-4】给定离散时间系统

$$\boldsymbol{x}(k+1)=0.3679\boldsymbol{x}(k)+0.6321\boldsymbol{u}(k)\qquad \boldsymbol{x}(0)=1$$

试决定最优控制率，使得下列性能指标达到极小。

$$J=\frac{1}{2}[\boldsymbol{x}(10)]^2+\frac{1}{2}\sum_{k=0}^{9}[\boldsymbol{x}^2(k)+\boldsymbol{u}^2(k)]$$

并求出 J 的最小值。

在本题中，$S=1$ 、$Q=1$ 、$R=1$ 。由式(8-35)Riccati 差分方程可得

$$P(k)=1+(0.3679)P(k+1)[1+(0.6321)(1)(0.6321)P(k+1)]^{-1}(0.6379)$$
$$=1+0.1354P(k+1)[1+0.3996P(k+1)]^{-1}$$

边界条件为

$$P(N)=P(10)=S=1$$

由 $k=9$ 到 $k=0$，求出 $P(k)$为

$$P(9)=1+0.1354\times1(1+0.3996\times1)^{-1}=1.0967$$
$$P(8)=1+0.1354\times1.0967(1+0.3996\times1.0967)^{-1}=1.1032$$
$$P(7)=1+0.1354\times1.1032(1+0.3996\times1.1032)^{-1}=1.1036$$
$$P(6)=1+0.1354\times1.1036(1+0.3996\times1.1036)^{-1}=1.1037$$
$$P(k)=1.1037 \qquad (k=5,4,3,2,1,0)$$

可以看出，$P(k)$迅速地趋近其稳态值。稳态值 P_{SS} 可由下式得到：

$$P_{SS}=1+0.1354P_{SS}(1+0.3996P_{SS})^{-1}$$

或

$$0.3996P_{SS}^2+0.4650P_{SS}-1=0$$

解得

$$P_{SS1}=1.1037 \qquad P_{SS2}=-2.2674$$

由于 $P(k)$必须是正定的，因此 $P(k)$的稳态值应为 1.1037 。

反馈增益可由式(8-40)得到。

$$K(k)=(1)(0.6321)(0.3679)^{-1}[P(k)-1]=1.7181[P(k)-1]$$

将 $P(k)$值代入上式，得

$$K(10)=1.7181(1-1)=0$$
$$K(9)=1.7181(1.0967-1)=0.1662$$
$$K(8)=1.7181(1.1032-1)=0.1773$$
$$K(7)=1.7181(1.1036-1)=0.1781$$
$$K(6)=K(5)=\cdots=K(0)=0.1781$$

由于

$$x(k+1)=0.3679x(k)+0.6321u(k)$$
$$=[0.3679-0.6321K(k)]x(k)$$

因此

$$x(1)=[0.3679-0.6321K(0)]x(0)$$

$=(0.3679-0.6321 \times 0.1781)\times 1 = 0.2553$

$\boldsymbol{x}(2)=(0.3679-0.6321 \times 0.1781)\times 0.2553 = 0.0652$

$\boldsymbol{x}(3)=(0.3679-0.6321 \times 0.1781)\times 0.0652 = 0.0166$

$\boldsymbol{x}(4)=(0.3679-0.6321 \times 0.1781)\times 0.0166 = 0.00424$

$\boldsymbol{k} = 5,6,\cdots,10$ 的 $\boldsymbol{x}(k)$值迅速趋近于零。

最优控制序列 $\boldsymbol{u}(k)$为

$\boldsymbol{u}(0)=-\boldsymbol{K}(0)\boldsymbol{x}(0)=-0.1781 \times 1 =-0.1781$

$\boldsymbol{u}(1)=-\boldsymbol{K}(1)\boldsymbol{x}(1)=-0.1781 \times 0.2553 =-0.0455$

$\boldsymbol{u}(2)=-\boldsymbol{K}(2)\boldsymbol{x}(2)=-0.1781 \times 0.0652 =-0.0116$

$\boldsymbol{u}(3)=-\boldsymbol{K}(3)\boldsymbol{x}(3)=-0.1781 \times 0.0166 =-0.00296$

$\boldsymbol{u}(4)=-\boldsymbol{K}(4)\boldsymbol{x}(4)=-0.1781 \times 0.00424 =-0.000756$

$\boldsymbol{u}(k)\approx 0 \qquad (k = 5,6,\cdots\cdots,10)$

最后，J 的最小值为

$$J_{\min}=\frac{1}{2}\boldsymbol{x}'(0)\boldsymbol{P}(0)\boldsymbol{x}(0)=\frac{1}{2}(1 \times 1.1037 \times 1)= 0.5518$$

MATLAB 程序 8-4 给出了本例题的解，包括 $\boldsymbol{K}(k)$、$\boldsymbol{x}(k)$、$\boldsymbol{u}(\mathrm{k})$及 $J_{\min}$，并点绘了 $\boldsymbol{P}(k)$、$\boldsymbol{K}(k)$对于 k 的曲线。图 8-4 为 $\boldsymbol{P}(k)$ 对 k 的图，图 8-5 为 $\boldsymbol{K}(k)$ 对 k 的图。

MATLB 程序 8-4 如下：

```
% MATLAB PROGRAM 8-4
% Design of discrete LQR system
A=[0.3679];B=[0.6321];S=[1];Q=[1];R=[1];P=[1];
P10=P;
K=inv(R)*B'*inv(A')*(P-Q);K10=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P9=P;
% Compute K,P until K=K0,P=P0
K=inv(R)*B'*inv(A')*(P-Q);K9=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P8=P;
K=inv(R)*B'*inv(A')*(P-Q);K8=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P7=P;
```

```
K=inv(R)*B'*inv(A')*(P-Q);K7=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P6=P;
K=inv(R)*B'*inv(A')*(P-Q);K6=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P5=P;
K=inv(R)*B'*inv(A')*(P-Q);K5=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P4=P;
K=inv(R)*B'*inv(A')*(P-Q);K4=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P3=P;
K=inv(R)*B'*inv(A')*(P-Q);K3=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P2=P;
K=inv(R)*B'*inv(A')*(P-Q);K2=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P1=P;
K=inv(R)*B'*inv(A')*(P-Q);K1=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A; P0=P;
K=inv(R)*B'*inv(A')*(P-Q);K0=K;
P=Q+A'*P*A-A'*P*B*inv(R+B'*P*B)*B'*P*A;
P=[P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10]'
P =
    1.1037
    1.1037
    1.1037
    1.1037
    1.1037
    1.1037
    1.1037
    1.1036
    1.1032
    1.0967
    1.0000
K=[K0 K1 K2 K3 K4 K5 K6 K7 K8 K9 K10]'
```

```
K =
    0.1781
    0.1781
    0.1781
    0.1781
    0.1781
    0.1781
    0.1781
    0.1781
    0.1773
    0.1662
         0
x=[1];x0=x;u=-K0*x;u0=u;
x=A*x+B*u;x1=x;  u=-K1*x;u1=u;
x=A*x+B*u;x2=x;  u=-K2*x;u2=u;
x=A*x+B*u;x3=x;  u=-K3*x;u3=u;
x=A*x+B*u;x4=x;  u=-K4*x;u4=u;
x=A*x+B*u;x5=x;  u=-K5*x;u5=u;
x=A*x+B*u;x6=x;  u=-K6*x;u6=u;
x=A*x+B*u;x7=x;  u=-K7*x;u7=u;
x=A*x+B*u;x8=x;  u=-K8*x;u8=u;
x=A*x+B*u;x9=x;  u=-K9*x;u9=u;
x=A*x+B*u;x10=x;  u=-K10*x;u10=u;
[x0 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10]'
ans =
    1.0000
    0.2553
    0.0652
    0.0166
    0.0042
```

```
    0.0011
    0.0003
    0.0001
    0.0000
    0.0000
    0.0000
[u0 u1 u2 u3 u4 u5 u6 u7 u8 u9 u10]'
ans =
   -0.1781
   -0.0455
   -0.0116
   -0.0030
   -0.0008
   -0.0002
    0.0000
    0.0000
    0.0000
    0.0000
         0
% Compute Jmin
Jmin=0.5*x0'*P0*x0
Jmin =
    0.5518
% Plot
k=0:10;
v=[0 10 0 1.2];
axis(v);
plot(k,P,'o')
grid
title('Plot of P(k) versus k');
```

```
xlabel('k')
ylabel('P(k)')
v=[0 10 0 0.2];
axis(v);
plot(k,K,'o')
grid
title('Plot of K(k) versus k');
xlabel('k')
ylabel('K(k)')
```

上述程序若用 for 循环，则可使程序缩短，改写后的程序如下：

```
A=[0.3679];B=[0.6321];S=[1];Q=[1];R=[1];
N=11;p(N)=S;x(1)=1;Pnext=S;
for i=N-1:-1:1
  P=Q+A'*Pnext*A-A'*Pnext*B*inv(R+B'*Pnext*B)*B'
  *Pnext*A;
  p(i)=P;Pnext=P;
end
for i=N:-1:1
  K=inv(R)*B'*inv(A')*(p(i)-Q);
  k(i)=K;
end
for i=1:N-1
  Xnext=(A-B*k(i))*x(i);
  x(i+1)=Xnext;
end
for i=1:N
  u(i)=-k(i)*x(i);
end
```

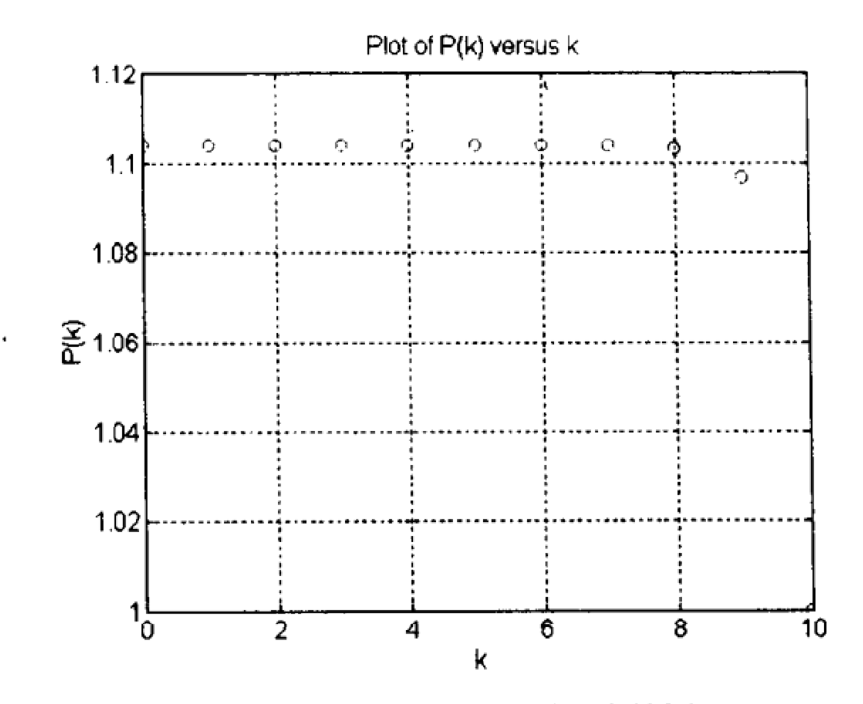


图 8-4　例 8-4 $\boldsymbol{P}(k)$对 k 点绘图

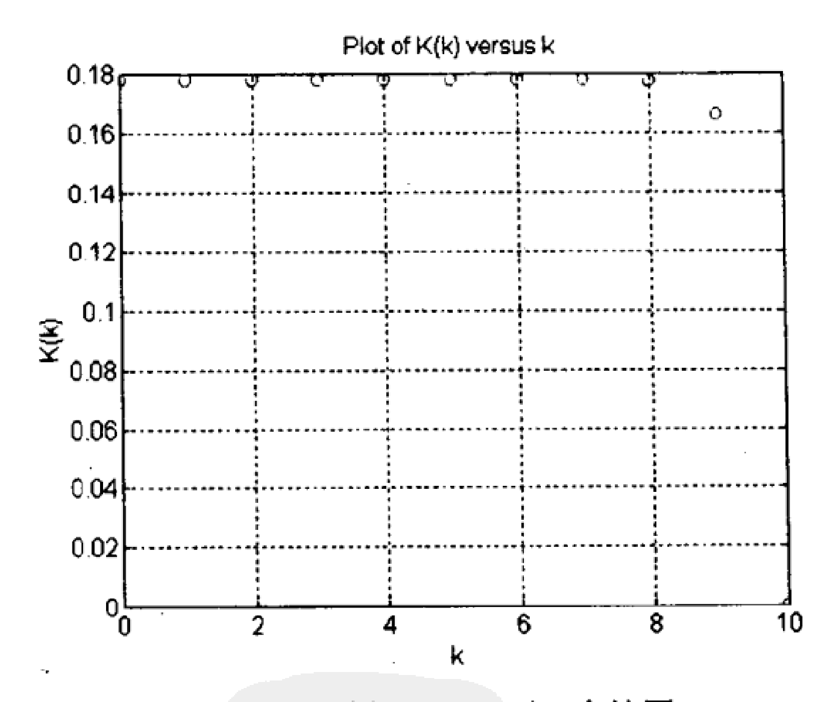


图 8-5　例 8-4 $\boldsymbol{K}(k)$对 k 点绘图

注意两个程序变量之间的对应关系。由于 MATLAB 矩阵的最小下标值为 1，因此对 $\boldsymbol{P}$ 来说，$\boldsymbol{P}(1)$对应程序 8-5 中的 P0，而 $\boldsymbol{P}(11)$对应 P10 。同理有 $\boldsymbol{k}(1)=$ k0, …, $\boldsymbol{k}(11)=$ k10 ； $\boldsymbol{x}(1)=$ x0,…,$\boldsymbol{x}(11)=$ x10 ； $\boldsymbol{u}(1)=$ u0, …, $\boldsymbol{u}(11)=$ u10 。而变量的打印

可实现如下：

```
M=[p' k' x' u']
M =
    1.1037    0.1781    1.0000   -0.1781
    1.1037    0.1781    0.2553   -0.0455
    1.1037    0.1781    0.0652   -0.0116
    1.1037    0.1781    0.0166   -0.0030
    1.1037    0.1781    0.0042   -0.0008
    1.1037    0.1781    0.0011   -0.0002
    1.1037    0.1781    0.0003    0.0000
    1.1036    0.1781    0.0001    0.0000
    1.1032    0.1773    0.0000    0.0000
    1.0967    0.1662    0.0000    0.0000
    1.0000         0    0.0000         0
```

M中的首列对应P0，…， P10；第二列至第四列分别对应k0，…，k10， x0，…， x10， u0，…， u10。

【例8-5】给定离散控制系统

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}\boldsymbol{u}(k)$$

式中

$$\boldsymbol{A}=\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}\qquad \boldsymbol{B}=\begin{bmatrix}1\\0\end{bmatrix}\qquad \boldsymbol{x}(0)=\begin{bmatrix}1\\0\end{bmatrix}$$

性能指标为

$$J=\frac{1}{2}\boldsymbol{x}'(8)\boldsymbol{S}\boldsymbol{x}(8)+\frac{1}{2}\sum_{k=0}^{7}\left[\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)\right]$$

式中

$$\boldsymbol{Q}=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\qquad \boldsymbol{R}=1\qquad \boldsymbol{S}=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$$

求系统最优控制序列 $\boldsymbol{u}(k)$。

解 参考式(8-35)有

$$\boldsymbol{P}(k)=\boldsymbol{Q}+\boldsymbol{A}'\boldsymbol{P}(k+1)[\boldsymbol{I}+\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{P}(k+1)]^{-1}\boldsymbol{A}$$

$$=\begin{bmatrix}1&0\\0&1\end{bmatrix}+\begin{bmatrix}1&1\\1&0\end{bmatrix}\begin{bmatrix}p_{11}(k+1)&p_{12}(k+1)\\p_{12}(k+1)&p_{22}(k+1)\end{bmatrix}\times$$

$$\left\{\begin{bmatrix}1&0\\0&1\end{bmatrix}+\begin{bmatrix}1&0\\0&0\end{bmatrix}\begin{bmatrix}p_{11}(k+1)&p_{12}(k+1)\\p_{12}(k+1)&p_{22}(k+1)\end{bmatrix}\right\}^{-1}\begin{bmatrix}1&1\\1&0\end{bmatrix}$$

$\boldsymbol{P}(k)$的边界条件为

$$\boldsymbol{P}(N)=\boldsymbol{P}(8)=\boldsymbol{S}=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

由$\boldsymbol{P}(7)$一直计算到$\boldsymbol{P}(0)$为

$$\boldsymbol{P}(7)=\begin{bmatrix}1&0\\0&1\end{bmatrix}+\begin{bmatrix}1&1\\1&0\end{bmatrix}\begin{bmatrix}1&0\\0&1\end{bmatrix}\left\{\begin{bmatrix}1&0\\0&1\end{bmatrix}+\begin{bmatrix}1&0\\0&0\end{bmatrix}\begin{bmatrix}1&0\\0&1\end{bmatrix}\right\}^{-1}\times$$

$$\begin{bmatrix}1&1\\1&0\end{bmatrix}$$

$$=\begin{bmatrix}2.5&0.5\\0.5&1.5\end{bmatrix}$$

$$\boldsymbol{P}(6)=\begin{bmatrix}1&0\\0&1\end{bmatrix}+\begin{bmatrix}1&1\\1&0\end{bmatrix}\begin{bmatrix}2.5&0.5\\0.5&1.5\end{bmatrix}\times$$

$$\left\{\begin{bmatrix}1&0\\0&1\end{bmatrix}+\begin{bmatrix}2.5&0.5\\0.5&1.5\end{bmatrix}\begin{bmatrix}1&0\\0&0\end{bmatrix}\right\}^{-1}\begin{bmatrix}1&1\\1&0\end{bmatrix}$$

$$=\begin{bmatrix}3.4286&0.8571\\0.8571&1.7143\end{bmatrix}$$

$\boldsymbol{P}(5)$、$\boldsymbol{P}(4)$、…、$\boldsymbol{P}(0)$也可以类似地计算出来，见表 8-1 。

下一步，求解反馈增益矩阵$\boldsymbol{K}(k)$。根据式(8-40)，$\boldsymbol{K}(k)$为

$$\boldsymbol{K}(k)=\boldsymbol{R}^{-1}\boldsymbol{B}'(\boldsymbol{A}')^{-1}[\boldsymbol{P}(k)-\boldsymbol{Q}]$$

$$=[1][1\quad 0]\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}^{-1}[\boldsymbol{P}(k)-\boldsymbol{Q}]$$

$$=[1\quad 0]\begin{bmatrix}p_{11}(k)-1 & p_{12}(k)\\ p_{12}(k) & p_{22}(k)-1\end{bmatrix}$$

$$=\begin{bmatrix}p_{12}(k) & p_{22}(k)-1\end{bmatrix}$$

这样

$$\boldsymbol{K}(8)=[P_{12}(8)\quad P_{22}(8)-1]=[0.0000\quad 0.0000]$$

$$\boldsymbol{K}(7)=[P_{12}(7)\quad P_{22}(7)-1]=[0.5000\quad 0.5000]$$

$\boldsymbol{K}(6)$、$\boldsymbol{K}(5)$、…、$\boldsymbol{K}(0)$也可以类似地计算出来，见表 8-1 。

再算出 $\boldsymbol{x}(k)$，由

$$\begin{aligned}\boldsymbol{x}(k+1) &= \boldsymbol{A}\,\boldsymbol{x}(k)+\boldsymbol{B}\,\boldsymbol{u}(k)\\ &=[\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}(k)]\boldsymbol{x}(k)\\ &=\begin{bmatrix}1-k_1(k) & 1-k_2(k)\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1(k)\\ x_2(k)\end{bmatrix}\end{aligned}$$

以及初始状态

$$\boldsymbol{x}(0)=[1\quad 0]'$$

可以算出 $\boldsymbol{x}(k)(k=1,2,\cdots,8)$为

$$\boldsymbol{x}(1)=\begin{bmatrix}1-1 & 1-0.7913\\ 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix}=\begin{bmatrix}0.0000\\ 1.0000\end{bmatrix}$$

$$\boldsymbol{x}(2)=\begin{bmatrix}1-0.9999 & 1-0.7913\\ 1 & 0\end{bmatrix}\begin{bmatrix}0.0000\\ 1.0000\end{bmatrix}=\begin{bmatrix}0.2087\\ 0.0000\end{bmatrix}$$

$\boldsymbol{x}(3)$、$\boldsymbol{x}(4)$、…、$\boldsymbol{x}(8)$也可以类似地算出，见表 8-1 。

最后，根据 $\boldsymbol{u}(k)=-\boldsymbol{K}(k)\boldsymbol{x}(k)$算出最优控制序列 $\boldsymbol{u}(k)$:

$$\boldsymbol{u}(0)=-\boldsymbol{K}(0)\boldsymbol{x}(0)=-[1\quad 0.7913]\begin{bmatrix}1\\ 0\end{bmatrix}=-1.0000$$

$$\boldsymbol{u}(1)=-\boldsymbol{K}(1)\boldsymbol{x}(1)=-[0.9999\quad 0.7913]\begin{bmatrix}0\\ 1\end{bmatrix}=-0.7913$$

$u(2)$、$u(3)$、…、$u(8)$也可以类似地算出来，见表 8-1。

表 8-1 $P(k)$，$K(k)$，$x(k)$，$u(k)$计算值

k	$P(k)$	$K(k)$	$x(k)$	$u(k)$
0	$\begin{bmatrix}3.7913 & 1.0000\\1.0000 & 1.7913\end{bmatrix}$	$\begin{bmatrix}1.0000 & 0.7913\end{bmatrix}$	$\begin{bmatrix}1.0000\\0.0000\end{bmatrix}$	−1.0000
1	$\begin{bmatrix}3.7911 & 0.9999\\0.9999 & 1.7913\end{bmatrix}$	$\begin{bmatrix}0.9999 & 0.7913\end{bmatrix}$	$\begin{bmatrix}0.0000\\1.0000\end{bmatrix}$	−0.7913
2	$\begin{bmatrix}3.7905 & 0.9997\\0.9997 & 1.7911\end{bmatrix}$	$\begin{bmatrix}0.9997 & 0.7911\end{bmatrix}$	$\begin{bmatrix}0.2087\\0.0000\end{bmatrix}$	−0.2087
3	$\begin{bmatrix}3.7877 & 0.9986\\0.9986 & 1.7905\end{bmatrix}$	$\begin{bmatrix}0.9986 & 0.7905\end{bmatrix}$	$\begin{bmatrix}0.0001\\0.2087\end{bmatrix}$	−0.1651
4	$\begin{bmatrix}3.7740 & 0.9932\\0.9932 & 1.7877\end{bmatrix}$	$\begin{bmatrix}0.9932 & 0.7877\end{bmatrix}$	$\begin{bmatrix}0.0437\\0.0001\end{bmatrix}$	−0.0435
5	$\begin{bmatrix}3.7097 & 0.9677\\0.9677 & 1.7742\end{bmatrix}$	$\begin{bmatrix}0.9677 & 0.7742\end{bmatrix}$	$\begin{bmatrix}0.0003\\0.0437\end{bmatrix}$	−0.0342
6	$\begin{bmatrix}3.4286 & 0.8571\\0.8571 & 1.7143\end{bmatrix}$	$\begin{bmatrix}0.8571 & 0.7143\end{bmatrix}$	$\begin{bmatrix}0.0099\\0.0003\end{bmatrix}$	−0.0087
7	$\begin{bmatrix}2.5000 & 0.5000\\0.5000 & 1.5000\end{bmatrix}$	$\begin{bmatrix}0.5000 & 0.5000\end{bmatrix}$	$\begin{bmatrix}0.0015\\0.0099\end{bmatrix}$	−0.0057
8	$\begin{bmatrix}1.0000 & 0.0000\\0.0000 & 1.0000\end{bmatrix}$	$\begin{bmatrix}0.0000 & 0.0000\end{bmatrix}$	$\begin{bmatrix}0.0057\\0.0015\end{bmatrix}$	−0.0000

正如前面所提到的，反馈增益矩阵$K(k)$除了最后几个值是时变的以外，其它的几乎都是常数。也就是说，如果步数由 8 改到 100 的话，$K(0)$、…、$K(93)$都是常数矩阵，只有$K(94)$、…、$K(100)$才是变化的。这一点很重要，因为若步数 N 充分大的话，反馈增益矩阵就变成常数矩阵，因此，设计者就可以用一个常数反馈增益矩阵去逼近时变的反馈增益矩阵了。

$J_{\min}$可由式(8-49)求得

$$J_{\min}=\frac{1}{2}x'(0)P(0)x(0)=\frac{1}{2}\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}3.7913 & 1.0000\\1.0000 & 1.7913\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}$$

$$=1.8956$$

下面的 MATLAB 程序 8-5 用来求出此例题的$P(k)$、$K(k)$、

$x(k)$、 $u(k)$及 J_{min} 等值。

```
% MATLAB PROGRAM 8-5
A=[1 1;1 0];B=[1;0];Q=[1 0;0 1];R=1;
S=[1 0;0 1];x0=[1;0];N=9;
p11(N)=1;p12(N)=0;p22(N)=1;x1(1)=1;x2(1)=0;
Pnext=S;
for i=N-1:-1:1
  P=Q+A'*Pnext*inv(eye(2)+B*inv(R)*B'*Pnext)*A;
  p11(i)=P(1,1);p12(i)=P(1,2);p22(i)=P(2,2);
  Pnext=P;
end
for i=N:-1:1
  K=inv(R)*B'*inv(A')*([p11(i) p12(i);p12(i) p22(i)]
  -Q);
  k1(i)=K(1);k2(i)=K(2);
end
for i=1:N-1
  Xnext=(A-B*[k1(i) k2(i)])*[x1(i);x2(i)];
  x1(i+1)=Xnext(1);x2(i+1)=Xnext(2);
end
for i=1:N
  u(i)=-[k1(i) k2(i)]*[x1(i);x2(i)];
end
P=[p11;p12;p12;p22]
P =
 Columns 1 through 7
  3.7913 3.7911 3.7905 3.7877 3.7740 3.7097 3.4286
  1.0000 0.9999 0.9997 0.9986 0.9932 0.9677 0.8571
  1.0000 0.9999 0.9997 0.9986 0.9932 0.9677 0.8571
  1.7913 1.7913 1.7911 1.7905 1.7877 1.7742 1.7143
```

```
 Columns 8 through 9
   2.5000   1.0000
   0.5000        0
   0.5000        0
   1.5000   1.0000
K=[k1;k2]'
K =
   1.0000    0.7913
   0.9999    0.7913
   0.9997    0.7911
   0.9986    0.7905
   0.9932    0.7877
   0.9677    0.7742
   0.8571    0.7143
   0.5000    0.5000
        0         0
x=[x1;x2]
x =
 Columns 1 through 7
   1.0000    0.0000    0.2087    0.0001    0.0437
   0.0003    0.0099
        0    1.0000    0.0000    0.2087    0.0001
   0.0437    0.0003
 Columns 8 through 9
   0.0015    0.0057
   0.0099    0.0015
u
u =
 Columns 1 through 7
  -1.0000   -0.7913   -0.2087   -0.1651   -0.0435
```

```
 -0.0342    -0.0087
 Columns 8 through 11
 -0.0057         0     0.0000          0
Jmin=0.5*x0'*[p11(1) p12(1);p12(1) p22(1)]*x0
Jmin =
     1.8956
```

8.4 离散系统的稳态二次型最优控制

8.4.1 离散系统稳态二次型最优控制问题的求解

考虑由下式定义的离散系统：

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\,\boldsymbol{x}(k)+\boldsymbol{B}\,\boldsymbol{u}(k) \tag{8-50}$$

由上一节的分析计算可以看出，若控制步数 N 是有限的，则反馈增益矩阵 $\boldsymbol{K}(k)$是时变的。现在将 N 由有限值改为无限值，即令 $N\to\infty$，则最优控制的解变成稳态解，原时变的反馈增益矩阵 $\boldsymbol{K}(k)$变成常数增益矩阵，并记为 $\boldsymbol{K}$ 。

当 $N=\infty$时，性能指标就改变为

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\ [\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)] \tag{8-51}$$

与式(8-15)相比，去掉了$(1/2)\boldsymbol{x}(N)\boldsymbol{S}\boldsymbol{x}(N)$项。这是因为若最优控制是稳定的，则 J 值将收敛到一个常数，而 $\boldsymbol{x}(\infty)$将变为 0 ，$(1/2)\boldsymbol{x}'(\infty)\boldsymbol{S}\boldsymbol{x}(\infty)=0$ 。

同时，$\boldsymbol{P}(k)$也变成常数矩阵 $\boldsymbol{P}$ ，由式(8-35)可得

$$\boldsymbol{P}=\boldsymbol{Q}+\boldsymbol{A}'\,[\boldsymbol{P}^{-1}+\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}']^{-1}\boldsymbol{A} \tag{8-52}$$

再由式(8-40)得

$$\begin{aligned}\boldsymbol{K}&=\boldsymbol{R}^{-1}\boldsymbol{B}'(\boldsymbol{A}')^{-1}(\boldsymbol{P}-\boldsymbol{Q})\\&=\boldsymbol{R}^{-1}\boldsymbol{B}'(\boldsymbol{P}^{-1}+\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}')\boldsymbol{A}\end{aligned} \tag{8-53}$$

对应的最优控制 $\boldsymbol{u}(k)$为

$$\begin{aligned}\boldsymbol{u}(k)&=-\boldsymbol{K}\,\boldsymbol{x}(k)\\&=-\boldsymbol{R}^{-1}\boldsymbol{B}'(\boldsymbol{P}^{-1}+\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}')\boldsymbol{A}\,\boldsymbol{x}(k)\end{aligned} \tag{8-54}$$

另外矩阵 $\boldsymbol{K}$ 也可由式(8-44)得到，即

$$K = (R+B'PB)^{-1} B'PA \tag{8-55}$$

对应 $u(k)$为

$$u(k) = -(R+B'PB)^{-1} B'PA\, x(k) \tag{8-56}$$

闭环系统的状态方程为

$$x(k+1) = [A-B(R+B'PB)^{-1} B'PA]x(k) \tag{8-57}$$

根据矩阵求逆引理得

$$(C+DE)^{-1} = C^{-1}-C^{-1}D(I+E\,C^{-1}D)^{-1}EC \tag{8-58}$$

并令上式中的 C 对应 I， D 对应 B， E 对应 $R^{-1}B'P$，则式(8-57)可写成

$$x(k+1) = (I+BR^{-1} B'P)^{-1}A\, x(k) \tag{8-59}$$

性能指标 $J_{\min}$ 没有变化，仍为

$$J_{\min} = \frac{1}{2} x'(0)P(0)x(0) \tag{8-60}$$

在实际应用中，常用常数矩阵 K 来代替时变矩阵 $K(k)$，在 N 为有限数时， K 与 $K(k)$只是在靠近终端时刻的值不同。

下面求解稳态离散 Riccati 方程。求解的方法有若干种，如差分方程法、 Schur 矢量法、迭代法等。这里采用的是一种迭代方法。由式(8-52)和式(8-58)的矩阵求逆引理，有

$$P = Q+A'PA-A'PB(R+B'PB)^{-1} B'PA \tag{8-61}$$

而迭代是由式(8-61)对应的非稳态方程开始计算。

$$P(k) = Q+A'P(k+1)A-A'P(k+1)B[R +B'P(k+1)B]^{-1}B'P(k+1)A \tag{8-62}$$

再将上式的 $P(k+1)$、 $P(k)$顺序倒过来有

$$P(k+1) = Q+A'P(k)A-A'P(k)B[R+B'P(k)B]^{-1} B'P(k)A \tag{8-63}$$

然后，由 $P(0) = 0$ 开始，将式(8-63)的迭代过程一直进行到相邻两次算出的 $P(k)$之差足够小时，便认为 $P(k)$已收敛到其稳态值 P。在计算中要注意，矩阵 P 应该是一个实对称正定矩阵。

8.4.2 MATLAB 实现

命令 dlqr 和 dlqry 可以直接求解稳态离散系统二次型调节器问题，以及相关的离散 Riccati 方程。这两个命令的格式如下：

[K, P, E] = dlqr(A, B, Q, R, N)

[K, P, E] = dlqry(A, B, C, D, Q, R)

式中， K 为离散最优反馈增益矩阵； P 为 Riccati 方程的唯一正定解； E 为 $\boldsymbol{A}-\boldsymbol{BK}$ 的特征值； N 为可选项，表示交叉乘积项的加权矩阵， dlqry 与 dlqr 的不同之处在于，在 dlqry 的目标函数中，用输出 $\boldsymbol{y}(k)$来代替状态 $\boldsymbol{x}(k)$，即目标函数为

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\left[\boldsymbol{y}'(k)\boldsymbol{Q}\boldsymbol{y}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)\right]$$

当带有可选项 $\boldsymbol{N}$ 时，目标函数变为

$$J=\frac{1}{2}\sum_{k=0}^{\infty}\left[\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)+2\boldsymbol{x}'(k)\boldsymbol{N}\,\boldsymbol{u}(k)\right]$$

命令 lqrd 用连续系统的性能指标函数来进行离散系统二次型调节器设计，即连续系统状态方程为

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}$$

性能指标为

$$J=\int_{0}^{\infty}(\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}'\boldsymbol{R}\boldsymbol{u})\mathrm{d}t$$

求离散系统最优反馈增益矩阵 $\boldsymbol{K}$，使得最优离散控制率 $\boldsymbol{u}(k)=-\boldsymbol{K}\boldsymbol{x}(k)$，使上述的连续系统性能指标 J 最小化。 lqrd 的命令格式如下：

[K, P, E] = lqrd(A, B, Q, R, Ts)

式中， A 、 B 为连续系统状态方程的系数矩阵； Q 、 R 为连续系统目标函数的加权矩阵， Ts 为采样周期。此命令先将 $\boldsymbol{A}$ 、 $\boldsymbol{B}$ 、 $\boldsymbol{Q}$ 、 $\boldsymbol{R}$ 以采样周期 T_S 及零阶保持器离散化，再求出离散系统最优反馈增益矩阵 $\boldsymbol{K}$，离散系统 Riccati 方程解 $\boldsymbol{P}$ 和 $\boldsymbol{E}$ 为离散状态方程

$$\boldsymbol{x}(k+1)=\boldsymbol{A}_{\mathrm{d}}\,\boldsymbol{x}(k)+\boldsymbol{B}_{\mathrm{d}}\,\boldsymbol{u}(k)$$

闭环后的系统矩阵 $\boldsymbol{A}_{\mathrm{d}}-\boldsymbol{B}_{\mathrm{d}}\boldsymbol{K}$ 的特征值。

【例 8-6 】给定系统

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\,\boldsymbol{x}(k)+\boldsymbol{B}\,\boldsymbol{u}(k)$$

式中

$$\boldsymbol{A}=\begin{bmatrix}0.3 & 0\\ 0 & 0.6\end{bmatrix}\qquad \boldsymbol{B}=\begin{bmatrix}1\\ 1\end{bmatrix}$$

性能指标为

$$J=\sum_{k=0}^{\infty}\ [\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)]$$

式中

$$\boldsymbol{Q}=\begin{bmatrix}1 & 0\\ 0 & 0.3\end{bmatrix}\qquad R=1$$

试求稳态最优反馈增益矩阵 $\boldsymbol{K}$ 和稳态离散 Riccati 方程的矩阵 $\boldsymbol{P}$ 的解。

解 求解此题的 MATLAB 程序 8-6 如下：

```
% MATLAB PROGRAM 8-6
A=[0.3 0;0 0.6];B=[1;1];Q=[1 0;0 0.3];R=1;
[K,P,E]=dlqr(A,B,Q,R)
K =
     0.1267     0.0983
P =
     1.0563    -0.0367
    -0.0367     0.4323
E =
     0.1390
     0.5360
```

【例 8-7】伺服系统设计系统结构图见图 8-6，采样周期 $T_S=0.1\mathrm{s}$。由图 8-6 可得到如下的系统方程：

$$x(k+1)=0.5x(k)+2\,u(k)$$

$$u(k)=k_1\,v(k)-k_2\,x(k)$$

$$v(k)=r(k)-y(k)+v(k-1)$$

$$y(k) = x(k)$$

式中，k_1为积分调节器的增益；k_2为反馈系数。要求决定k_1、k_2两个参数，使得系统稳定，并观察系统的单位阶跃响应。

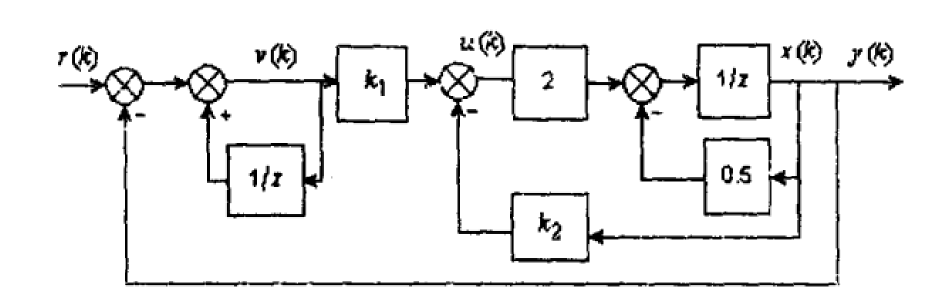


图 8-6　例 8-7 系统结构图

解　由系统方程可得

$$v(k+1) = r(k+1) - y(k+1) + v(k)$$
$$= -0.5x(k) + v(k) - 2u(k) + r(k+1)$$

与$x(k+1)$组成矩阵有

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(k+1) \tag{8-64}$$

对于k =∞，有

$$\begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} x(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \end{bmatrix} u(\infty) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r(\infty) \tag{8-65}$$

当系统为阶跃输入时，$r(k+1) = r(\infty) = r$。

令$x_e(k) = x(k) - x(\infty)$，$v_e(k) = v(k) - v(\infty)$，$u_e(k) = u(k) - u(\infty)$。将式(8-64)减去式(8-65)，得

$$\begin{bmatrix} x_e(k+1) \\ v_e(k+1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} x_e(k) \\ v_e(k) \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \end{bmatrix} u_e(k) \tag{8-66}$$

再令$x_1(k) = x_e(k)$，　$x_2(k) = v_e(k)$，　$w(k) = u_e(k)$，并由$u_e(k) = k_1 v_e(k) - k_2 x_e(k)$，则式(8-66)可写成

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \end{bmatrix} w(k)$$

式中

$$w(k) = -\begin{bmatrix} k_2 & -k_1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

写成状态方程形式为

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}w(k) \qquad w(k) = -\boldsymbol{K}\boldsymbol{x}(k)$$

式中

$$\boldsymbol{x}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \quad \boldsymbol{A} = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 1 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 2 \\ -2 \end{bmatrix} \quad \boldsymbol{K} = \begin{bmatrix} k_2 & -k_1 \end{bmatrix}$$

对于这个系统，采用二次型最优控制的方法来设计，由此得到的系统总是稳定的。

设性能指标如下：

$$J = \sum_{k=0}^{\infty} [\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k) + w'(k)\boldsymbol{R}\, w(k)]$$

式中的 $\boldsymbol{Q}$ 、 $\boldsymbol{R}$ 选择如下：

$$\boldsymbol{Q} = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \quad R = 1$$

注意，这只是 $\boldsymbol{Q}$ 、 R 的一种选择。若选择其它的正定的 $\boldsymbol{Q}$ 、 $\boldsymbol{R}$ ，所得到的系统也是稳定的。但对应的 R_1 、 R_2 参数、系统响应会有所不同。

MATLAB 程序 8-7 用来决定矩阵 $\boldsymbol{K}$ 。

```
% MATLAB PROGRAM 8-7
% Servo system design
A=[0.5 0;-0.5 1];B=[2;-1];Q=[100 0;0 1];R=1;
[K,P,E]=dlqr(A,B,Q,R)
K =
     0.2614    -0.0481
```

```
P =
   100.8917    -3.3053
    -3.3053    14.1731
E =
     0.0012
     0.9277
k1=-K(2);k2=K(1);
AA=[0.5-2*k2 2*k1;-0.5+2*k2 1-2*k1];
BB=[0;1];CC=[1 0];DD=0;FF=[0 1];
[num,den]=ss2tf(AA,BB,CC,DD);
r=ones(1,101);
axis([0 100 0 1.4]);k=0:100;
y=filter(num,den,r);
plot(k,y,'o',k,y,'-')
grid
title('Output y(k) to Unit-Step Input')
xlabel('k')
ylabel('y(k)')
[numv,denv]=ss2tf(AA,BB,FF,DD);
axis([0 100 0 12]);k=0:100;
v=filter(numv,denv,r);
plot(k,v,'o',k,v,'-')
grid
title('Output v(k) of Integrator')
xlabel('k')
ylabel('v(k)')
```

下面求该系统的单位阶跃响应。将下式

$$u(k)=k_1 v(k)-k_2 x(k)=\begin{bmatrix}-k_2 & k_1\end{bmatrix}\begin{bmatrix}x(k)\\ v(k)\end{bmatrix}$$

代入式(8-64)，并考虑到阶跃响应，令 $r=r(k)$，有

$$\begin{bmatrix} x(k+1) \\ v(k+1) \end{bmatrix} = \begin{bmatrix} 0.5-2k_2 & 2k_1 \\ -0.5+2k_2 & 1-2k_1 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \tag{8-67}$$

$$y(k)=x(k)=\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \end{bmatrix} + [\,0\,]\,r \tag{8-68}$$

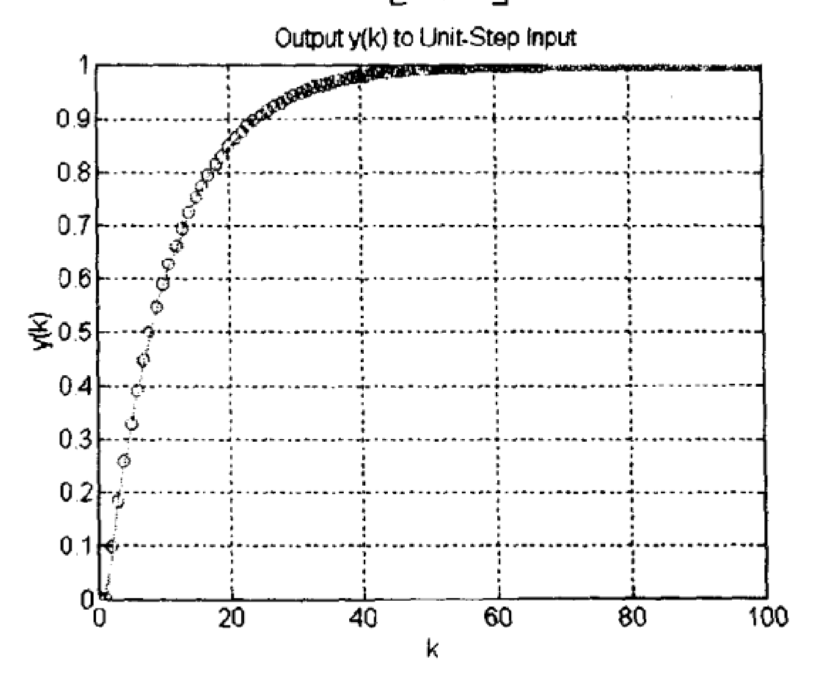


图 8-7　例 8-7 $y(k)$的单位阶跃响应

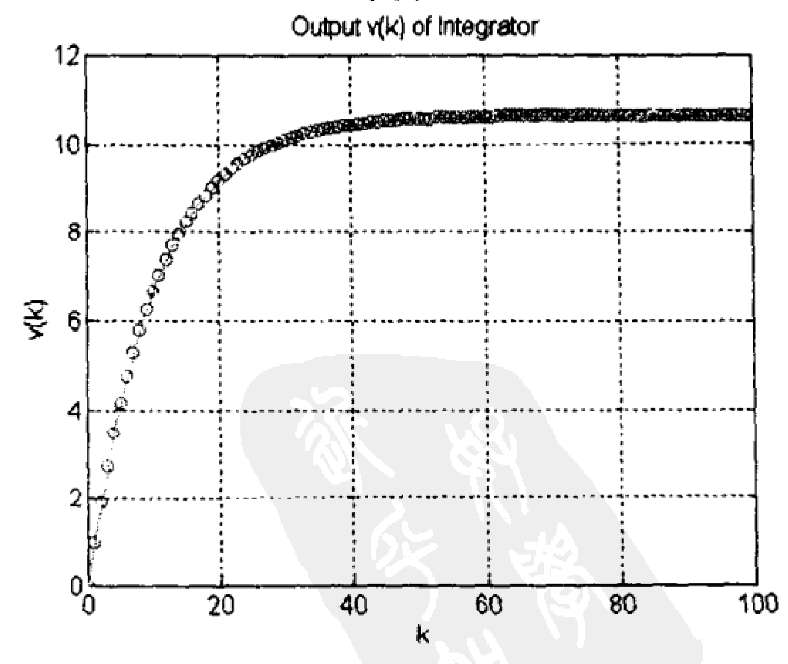


图 8-8　例 8-8 $v(k)$的单位阶跃响应

先将上面的离散状态方程用 MATLAB 命令 ss2tf 转换成脉冲传递函数。

[num, den]= ss2tf (AA, BB, CC, DD)

式中， AA =[0.5−2k_2 2k_1；−0.5+2k_2 1−2k_1]；

BB =[0； 1]； CC =[1 0]； DD =[0]。再用 MATLAB 命令 filter 得到单位阶跃响应

y = filter(num, den, r)

式中， r 为单位阶跃函数。

另外，若想得到 $v(k)$的响应，可根据

$$v(k)=[\,0\quad 1\,]\begin{bmatrix}x(k)\\ v(k)\end{bmatrix}=\boldsymbol{FF}\begin{bmatrix}x(k)\\ v(k)\end{bmatrix}$$

式中， $\boldsymbol{FF}$ =[0 1]。再用以下命令求得 $v(k)$

[numv, denv]= ss2tf(AA, BB, FF, DD)

v = filter(numv, denv, r)

用下面的 MATLAB 程序可求得 $y(k)$和 $v(k)$的单位阶跃响应曲线，见图 8-7 、图 8-8 。

```
k1=-K(2);k2=K(1);
AA=[0.5-2*k2 2*k1;-0.5+2*k2 1-2*k1];
BB=[0;1];CC=[1 0];DD=0;FF=[0 1];
[num,den]=ss2tf(AA,BB,CC,DD);
r=ones(1,101);
axis([0 100 0 1.4]);k=0:100;
y=filter(num,den,r);
plot(k,y,'o',k,y,'-')
grid
title('Output y(k) to Unit-Step Input')
xlabel('k')
ylabel('y(k)')
[numv,denv]=ss2tf(AA,BB,FF,DD);
```

```
axis([0 100 0 12]);k=0:100;
v=filter(numv,denv,r);
plot(k,v,'o',k,v,'-')
grid
title('Output v(k) of Integrator')
xlabel('k')
ylabel('v(k)'
```

8.5 最少能量控制问题

8.5.1 伪逆

由于伪逆在最少能量控制系统中起到了重要作用，因此首先回顾有关伪逆的概念。

伪逆是逆的概念的推广。它对于代数方程中未知变量的个数与线性无关的方程的个数不同时的求解很有用。

考虑一个线性代数方程

$$x_1+5x_2 = 2$$

由于只有一个方程，却有两个变量，因此方程有无数组解。其解为 x_1， x_2 平面上的一条直线上的所有点，但是若要得到最接近原点的点(解)，则解是唯一的。

考虑代数方程

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{8-69}$$

式中，$\boldsymbol{A}$ 为 $n \times m$ 维矩阵；$\boldsymbol{x}$ 为 m 维矢量；$\boldsymbol{b}$ 为 n 维矢量。设 $m > n$，即未知变量的个数大于方程的个数，则方程有无数组解。现在是要找出最接近原点的解，即最小范数解，则它是唯一的。令这个最小范数解为 $\boldsymbol{x}^0$。即 $\boldsymbol{x}^0$ 满足以下条件：$\boldsymbol{Ax}^0 = \boldsymbol{b}$，且对所有满足 $\boldsymbol{Ax} = \boldsymbol{b}$ 的 $\boldsymbol{x}$，有$\|\boldsymbol{x}^0\| \leqslant \|\boldsymbol{x}\|$。这表示在 $\boldsymbol{x}$ 对应的 m 维空间中，$\boldsymbol{x}^0$ 是所有满足式(8-69)的解中最靠近原点的。$\boldsymbol{x}^0$ 由下式给出：

$$\boldsymbol{x}^0 = \boldsymbol{A}^{\mathrm{RM}}\boldsymbol{b}$$

式中，$\boldsymbol{A}^{\mathrm{RM}} = \boldsymbol{A}'(\boldsymbol{AA}')^{-1}$。其证明如下：

$$\|x\| = \|x-x^0+x^0\| = \|x^0\|+\|x-x^0\|+2(x^0)'(x-x^0)$$

上式的最后一项为

$$\begin{aligned} 2(x^0)'(x-x^0) &= 2[A'(AA')^{-1}b]'[x-A'(AA')^{-1}b] \\ &= 2b'(AA')^{-1}A[x-A'(AA')^{-1}b] \\ &= 2b'(AA')^{-1}[Ax-(AA')(AA')^{-1}b] \\ &= 2b'(AA')^{-1}[b-b] = 0 \end{aligned}$$

因此

$$\|x\| = \|x^0\|+\|x-x^0\|$$

或 $\|x\|-\|x^0\| = \|x-x^0\|$

由于$\|x-x^0\| \geqslant 0$

则有$\|x\| \geqslant \|x^0\|$

这个使$\|x^0\|$最小的矩阵 $A^{RM} = A'(AA)^{-1}$ 称为右伪逆矩阵，其维数为 $m \times n$ 维。而且 A^{RM} 确实具有逆矩阵的性质，即

$$AA^{RM} = A[A'(AA')^{-1}] = AA'(AA')^{-1} = I_n$$

下面再看一下最小化$\|Ax-b\|$的解。考虑下面的线性代数方程组

$$Ax = b \tag{8-70}$$

式中，A 为 $n \times m$ 维矩阵；x 为 m 维矢量；b 为 n 维矢量。设 $m < n$，即未知变量的个数小于方程的个数。一般来说，这种方程组可能无解。如果方程组无解，则希望能够找到可以使范数 $\|Ax-b\|$ 最小化的解。令此解为 x^0，即此解满足以下条件：

$$\|Ax-b\| \geqslant \|Ax^0-b\| \qquad \forall x \in C^m$$

这里，x^0 不是传统意义上的解，因为它并不满足代数方程组 $Ax = b$。因此 x^0 被称为可使$\|Ax-b\|$最小化的近似解。x^0 由下式给出：

$$x^0 = A^{LM}b = (A'A)^{-1}A'b$$

式中，$A^{LM} = (A'A)^{-1}A'$。其证明如下：

$$\begin{aligned} \|Ax-b\| &= \|A(x-x^0)+Ax^0-b\| \\ &= \|A(x-x^0)\|+\|Ax^0-b\|+2[A(x-x^0)]'(Ax^0-b) \end{aligned}$$

上式的最后一项为

$$2[A(x-x^0)]'(Ax^0-b) = 2(x-x^0)'A'[A(A'A)^{-1}A'-I_n]b$$
$$= 2(x-x^0)'[A'A(A'A)^{-1}A'-A']b$$
$$= 2(x-x^0)'(A'-A')b = 0$$

因此

$$\|Ax-b\| = \|A(x-x^0)\| + \|Ax^0-b\|$$

由于$\|A(x-x^0)\| \geqslant 0$，则有

$$\|Ax-b\| \geqslant \|Ax^0-b\|$$

这个使$\| Ax^0-b \|$为最小的矩阵 $A^{LM}=(A'A)^{-1}A'$ 称为左伪逆矩阵，其维数为 $m \times n$ 维。而且 A^{LM} 确实也具有逆矩阵的性质。

$$A^{LM}A=(A'A)^{-1}A'A = I_m$$

8.5.2 伪逆的 MATLAB 实现

命令 pinv 用来计算矩阵的伪逆。其命令格式为

L = pinv(A)

式中，A 为 $n \times m$ 维矩阵；L 与 A' 具有相同的维数。若 $n < m$，则 L 为 A 的右伪逆矩阵，即 $L = A^{RM} = A'(AA')^{-1}$。若 $n > m$，则 L 为 A 的左伪逆矩阵，即 $L = A^{LM}=(A'A)^{-1}A'$。

【例 8-8】 给定线性代数方程组

$$Ax = b$$

式中

$$A=\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix} \quad x=\begin{bmatrix}x_1\\x_2\\x_3\end{bmatrix} \quad b=\begin{bmatrix}10\\20\end{bmatrix}$$

求此方程的最小范数解 x^0。

解 $x^0 = A^{RM}b = A'(AA')^{-1}b$

$$=\begin{bmatrix}1 & 4\\2 & 5\\3 & 6\end{bmatrix}\left\{\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\end{bmatrix}\begin{bmatrix}1 & 4\\2 & 5\\3 & 6\end{bmatrix}\right\}^{-1}\begin{bmatrix}10\\20\end{bmatrix}$$

$$
=\begin{bmatrix}-0.5556\\1.1111\\2.7778\end{bmatrix}
$$

下面是解此题的 MATLAB 程序 8-8。

```
% MATLAB PROGRAM 8-8
% Right pseudoinverse matrix
A=[1 2 3;4 5 6];b=[10;20];
ARM=pinv(A)
ARM =
   -0.9444    0.4444
   -0.1111    0.1111
    0.7222   -0.2222
x0=ARM*b
x0 =
   -0.5556
    1.1111
    2.7778
A*ARM
ans =
    1.0000    0.0000
    0.0000    1.0000
ARM*A
ans =
    0.8333    0.3333   -0.1667
    0.3333    0.3333    0.3333
   -0.1667    0.3333    0.8333
```

注意，上面程序中 A*ARM 的维数(2 × 2)与 ARM*A 的维数(3 × 3)的不同。

【例 8-9】给定线性代数方程组

$$
\boldsymbol{Ax}=\boldsymbol{b}
$$

式中

$$A=\begin{bmatrix}1 & 1\\1 & 2\\1 & 4\end{bmatrix} \qquad x=\begin{bmatrix}x_1\\x_2\end{bmatrix} \qquad b=\begin{bmatrix}1\\2\\3\end{bmatrix}$$

求此方程组的满足最小化$\|Ax-b\|$ 的最小范数解。

解 $x^0 = A^{LM}b = (A'A)^{-1}A'b$

$$=\left\{\begin{bmatrix}1 & 1 & 1\\1 & 2 & 4\end{bmatrix}\begin{bmatrix}1 & 1\\1 & 2\\1 & 4\end{bmatrix}\right\}^{-1}\begin{bmatrix}1 & 1 & 1\\1 & 2 & 4\end{bmatrix}\begin{bmatrix}1\\2\\3\end{bmatrix}=\begin{bmatrix}0.5000\\0.6429\end{bmatrix}$$

下面是解此题的 MATLAB 程序 8-9 。

```
% MATLAB PROGRAM 8-9
% Left pseudoinverse matrix
A=[1 1;1 2;1 4];b=[1;2;3];
ALM=pinv(A)
ALM =
    1.0000    0.5000   -0.5000
   -0.2857   -0.0714    0.3571
x0=ALM*b
x0 =
    0.5000
    0.6429
ALM*A
ans =
    1.0000    0.0000
    0.0000    1.0000
A*ALM
ans =
    0.7143    0.4286   -0.1429
    0.4286    0.3571    0.2143
```

-0.1429　　0.2143　　0.9286

注意，上面程序中 ALM*A 的维数(2 × 2)与 A*ALM 的维数(3 × 3)的不同。

8.5.3　最少能量控制问题的讨论

如果一个 n 阶单输入单输出离散系统是完全能控的，则可以适当地选取控制信号，使得系统可以在最多 n 个采样周期内，从任意的初始状态转移到期望的目标状态，只要控制矢量不受约束即可。如果允许的采样周期个数 $N > n$，则可得到额外的自由度，使得系统能满足其它约束条件。

控制系统所需的控制能量的多少取决于控制所允许的采样周期数，若允许的采样周期数为系统的阶数 n，则时间最优控制序列 $u(0)$、$u(1)$、…、$u(n-1)$是唯一的。但是，如果允许的采样周期数为 N，$N > n$，则可能有多种控制序列。每种控制序列要求的控制能量会有所不同。在许多工业应用中，若有许多种控制序列可供选择的话，经常要求用最少的控制能量去完成控制任务。

这样，对控制序列的要求是：在 N 个采样周期内，用最少的控制能量，使系统状态由任意初始状态转移到期望的目标状态。在这里，假设期望的目标状态为状态空间的零点。

考虑下列离散系统：

$$\boldsymbol{x}(k+1)=\boldsymbol{A}\boldsymbol{x}(k)+\boldsymbol{B}u(k) \tag{8-71}$$

式中，$\boldsymbol{x}(k)$为 n 维状态矢量；$u(k)$为标量控制信号；$\boldsymbol{A}$ 为 $n \times m$ 维非奇异矩阵；$\boldsymbol{B}$ 为 n 维矢量。

下面，就是要求出能在 $N(N > n)$个采样周期内使得系统状态由任意初始状态转移到状态空间原点，其使能量控制性能指标

$$J=\frac{1}{2}\sum_{k=0}^{N-1} u^2(k)$$

为最小的控制序列。

设系统是完全可控的，式(8-71)的 $\boldsymbol{x}(N)$可由下式给出：

$$x(N)=A^N x(0)+A^{N-1}Bu(0)+A^{N-2}Bu(1)+\cdots$$
$$+ABu(N-2)+Bu(N-1)$$

将 $x(N)=0$ 代入上式，则有

$$x(0)=-A^{-1}B\,u(0)-A^{-2}Bu(1)-\cdots-A^{-N+1}Bu(N-2)-A^{-N}Bu(N-1) \tag{8-72}$$

令 $f_i=A^{-i}B$，则上式变成

$$x(0)=-f_1\,u(0)-f_2\,u\,(2)-\cdots-f_{N-1}\,u(N-2)-f_N\quad u(N-1) \tag{8-73}$$

由于系统是完全可控的，则矢量 f_1、f_2、…、f_n 是线性无关的，而其余的 $N-n$ 个矢量可由这 n 个线性无关矢量线性表达。这样，式(8-73)可重写为

$$x(0)=-Fu \tag{8-74}$$

式中

$$F=[f_1\vdots\ f_2\vdots\ \cdots\vdots\ f_N]\qquad u=\begin{bmatrix}u(0)\\u(1)\\\vdots\\u(N-1)\end{bmatrix}$$

下面要找出能满足式(8-74)，且又可使控制能量最少的控制序列。

矩阵 F 是一个 $n\times N$ 维矩阵，其秩为 n。由于 F 不是一个方阵，F 的逆没有定义。而且由于 $N>n$，则未知控制信号 $u(0)$、$u(1)$、…、$u(N-1)$个数大于方程的个数。因此，这个方程组有无数组解。但是，对于这个系统的N个未知控制变量$u(0)$、$u(1)$、…、$u(N-1)$来说，是有最小范数

$$\frac{1}{2}\sum_{k=0}^{N-1}u^2(k)=\min$$

来约束的。因此，有唯一解，即最少能量解。

由于在解此问题时，未知变量的个数大于方程的个数。因此，应该采用右伪逆阵的方法来解。右伪逆阵 F^{RM} 为

$$F^{RM}=F'(FF')^{-1} \tag{8-75}$$

这样，最少能量控制序列 u (0)、$u(1)$、…、$u(N-1)$就由下

式得到：

$$u=-F^{RM}x(0)=-F'(FF')^{-1}x(0) \tag{8-76}$$

或

$$\begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix} = -F'(FF')^{-1}x(0) \tag{8-77}$$

【例 8-10】 给定系统

$$x(k+1)=Ax(k)+Bu(k) \tag{8-78}$$

式中

$$A=\begin{bmatrix} 1 & 0.6321 \\ 0 & 0.3679 \end{bmatrix} \quad B=\begin{bmatrix} 0.3679 \\ 0.6321 \end{bmatrix} \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}=\begin{bmatrix} 5 \\ -5 \end{bmatrix}$$

设采样周期为 1s 。要求在 3 个采样周期内将初始状态转移到原点。这有无数组控制序列。但要求的是使控制能量最少，即令性能指标

$$J=\frac{1}{2}\sum_{k=0}^{N-1} u^2(k)$$

为最小的控制序列。

由式(8-73)，初始状态 $x(0)$可表示为

$$x(0)=-f_1 u(0)-f_2 u(1)-f_3 u(2)$$

式中

$$f_1=A^{-1}B=\begin{bmatrix} -0.7181 \\ 1.7181 \end{bmatrix} \qquad f_2=A^{-2}B=\begin{bmatrix} -3.6701 \\ 4.6701 \end{bmatrix}$$

$$f_3=A^{-3}B=\begin{bmatrix} -11.6939 \\ 12.6939 \end{bmatrix}$$

因此有

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}=-\begin{bmatrix} -0.7181 \\ 1.7181 \end{bmatrix}u(0)-\begin{bmatrix} -3.6701 \\ 4.6701 \end{bmatrix}u(1)$$

$$-\begin{bmatrix} -11.6939 \\ 12.6939 \end{bmatrix} u(2)$$

或

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = -\begin{bmatrix} -0.7181 & -3.6701 & -11.6939 \\ 1.7181 & 4.6701 & 12.6939 \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ u(2) \end{bmatrix} \tag{8-79}$$

用右伪逆阵，可以得到式(8-79)的最小范数解

$$\begin{bmatrix} u(0) \\ u(1) \\ u(2) \end{bmatrix} = -\boldsymbol{F}^{\mathrm{RM}} \boldsymbol{x}(0) = -\boldsymbol{F}'(\boldsymbol{F}\boldsymbol{F}')^{-1} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$$

式中

$$\boldsymbol{F} = \begin{bmatrix} -0.7181 & -3.6701 & -11.6939 \\ 1.7181 & 4.6701 & 12.6939 \end{bmatrix}$$

右伪逆阵 $\boldsymbol{F}^{\mathrm{RM}}$ 为

$$\begin{aligned} \boldsymbol{F}^{\mathrm{RM}} &= \boldsymbol{F}'(\boldsymbol{F}\boldsymbol{F}')^{-1} \\ &= \begin{bmatrix} -0.7181 & 1.7181 \\ -3.6701 & 4.6701 \\ -11.6939 & 12.6939 \end{bmatrix} \begin{bmatrix} 150.7326 & -166.8147 \\ -166.8147 & 185.8968 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 0.7910 & 0.7191 \\ 0.5000 & 0.4738 \\ -0.2910 & -0.1929 \end{bmatrix} \end{aligned}$$

控制序列为

$$\begin{bmatrix} u(0) \\ u(1) \\ u(2) \end{bmatrix} = -\begin{bmatrix} 0.7910 & 0.7191 \\ 0.5000 & 0.4738 \\ -0.2910 & -0.1929 \end{bmatrix} \begin{bmatrix} 5 \\ -5 \end{bmatrix} = \begin{bmatrix} -0.3598 \\ -0.1310 \\ -0.4908 \end{bmatrix} \tag{8-80}$$

由式(8-80)给出的控制序列将在三个采样周期内使系统状态由

初始状态转移到状态空间的零点，且总的控制能量为最少。在这种情况下，状态转移的过程为

$$\begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} 1 & 0.6321 \\ 0 & 0.3679 \end{bmatrix} \begin{bmatrix} 5 \\ -5 \end{bmatrix} + \begin{bmatrix} 0.3679 \\ 0.6321 \end{bmatrix} [-0.3598]$$

$$= \begin{bmatrix} 1.7071 \\ -2.0669 \end{bmatrix}$$

$$\begin{bmatrix} x_1(2) \\ x_2(2) \end{bmatrix} = \begin{bmatrix} 1 & 0.6321 \\ 0 & 0.3679 \end{bmatrix} \begin{bmatrix} 1.7071 \\ -2.0669 \end{bmatrix} + \begin{bmatrix} 0.3679 \\ 0.6321 \end{bmatrix} [-0.1310]$$

$$= \begin{bmatrix} 0.3524 \\ -0.8432 \end{bmatrix}$$

$$\begin{bmatrix} x_1(3) \\ x_2(3) \end{bmatrix} = \begin{bmatrix} 1 & 0.6321 \\ 0 & 0.3679 \end{bmatrix} \begin{bmatrix} 0.3524 \\ -0.8432 \end{bmatrix} + \begin{bmatrix} 0.3679 \\ 0.6321 \end{bmatrix} [-0.4908]$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

所要求的最少控制能量为

$$J_{\min} = \frac{1}{2} \sum_{k=0}^{N-1} u^2(k) = \frac{1}{2} [u^2(0) + u^2(1) + u^2(2)]$$

$$= \frac{1}{2} [(-0.3598)^2 + (-0.1310)^2 + (-0.4908)^2]$$

$$= 0.1937$$

MATLAB 程序 8-10 给出了求解此题的计算过程。

```
% MATLAB PROGRAM 8-10
% Minimum energy control
A=[1 0.6321;0 0.3679];B=[0.3679;0.6321];x0=[5;-5];
f1=inv(A)*B;
f2=inv(A)^2*B;
f3=inv(A)^3*B;
```

```
F=[f1 f2 f3];
FRM=pinv(F)
FRM =
     0.7910      0.7191
     0.5000      0.4738
    -0.2910     -0.1929
uopt=-FRM*x0
uopt =
    -0.3598
    -0.1310
     0.4908
u0=uopt(1);u1=uopt(2);u2=uopt(3);
Jmin=0.5*uopt'*uopt
Jmin =
     0.1937
x1=A*x0+B*u0
x1 =
     1.7071
    -2.0669
x2=A*x1+B*u1
x2 =
     0.3524
    -0.8432
x3=A*x2+B*u2
x3 =
   1.0e-015 *
  -0.4441
  -0.0555
```

8.6 最优观测器设计

估计理论和 Kalman-Bucy 滤波器在控制领域和通信领域都有广泛的应用。二次型最优控制要求得到所有状态反馈。而实际上，在得到状态反馈时，也会同时受到量测噪声的干扰。最优估计问题就是要从带有量测噪声的检测状态中获得状态的最优估计。我们将首先导出有关的公式，求出解，并说明如何用 MATLAB 来实现和仿真这种滤波器。

8.6.1 公式与求解

考虑如下系统：

$$\dot{x} = Ax+Bu+Fw \qquad (8\text{-}81)$$
$$y = Cx+v$$

式中，w 为随机噪声干扰输入；v 为由传感器带来的随机量测噪声。这里需要对噪声过程的系统特性作出假定，即两个噪声都是具有零均值和已知方差及协方差的高斯白噪声过程。这些统计特性如下：

$$\mathrm{E}\{w(t)\}=0 \quad \mathrm{E}\{v(t)\}=0 \qquad (8\text{-}82)$$
$$\mathrm{E}\{w(t)w'(t+\tau)\}=Q_0\,\delta(t-\tau)$$
$$\mathrm{E}\{v(t)v'(t+\tau)\}=R_0\,\delta(t-\tau)$$
$$\mathrm{E}\{w(t)v'(t+\tau)\}=0 \qquad \forall\, t,\ \tau$$

现在需要做的是求取带有噪声的量测的估计，使得误差的方差最小。

令 $\hat{x}(t)$ 为估计值、$\tilde{x}(t)$ 为误差，性能指标由下式给出：

$$J=\mathrm{E}\left\{\tilde{x}'(t)\tilde{x}(t)\right\} \qquad (8\text{-}83)$$

式中，$\tilde{x}(t)=x(t)-\hat{x}(t)$。为了得到一个渐近稳定的最小方差滤波器，先做下列假设，即$(C,\ A)$是完全可观测的，矩阵 R_0 是正定的，存在 H_0 使得 $H_0H_0'=Q_0$，且$(A,\ H_0)$是可稳定的。在上述假设条件下，最优估计器由下式给出：

$$\dot{\hat{x}}=A\hat{x}+Bu+L(y-C\hat{x}) \qquad (8\text{-}84)$$
$$L=P_0C'R_0^{-1}$$

式中，P_0 由下式得出：

$$AP_0+P_0A'+FQ_0F'-P_0C'R_0^{-1}CP_0=0 \qquad (8\text{-}85)$$

可以证明式(8-85)Riccati 方程的解 P_0 就是估计误差的协方差，而此协方差的迹即为误差方差。

$$\mathrm{tr}\,P_0 = \mathrm{tr}\,\mathrm{E}\left\{\tilde{x}(t)\tilde{x}'(t)\right\} = \mathrm{E}\left\{\tilde{x}'(t)\tilde{x}(t)\right\} \tag{8-86}$$

因此，P_0 的迹就表征了滤波器性能的优劣。

若对前面的假设条件再放宽一些，即系统的噪声过程之间是相关的，就可以得到更一般形式的滤波器。设噪声扰动输入与量测噪声是相关的，则可以得到一个协方差阵为

$$\mathrm{E}\left\{\begin{bmatrix} w(t) \\ v(t) \end{bmatrix}\begin{bmatrix} w'(t) & v'(t) \end{bmatrix}\right\} = \begin{bmatrix} Q_0 & N_0 \\ N_0' & R_0 \end{bmatrix}\delta(t-\tau) \tag{8-87}$$

在这种情况下，最优观测器求解方程就变成

$$L = (P_0C' + N_0)R_0^{-1}$$

$$AP_0 + P_0A' + FQ_0F' - (P_0C' + N_0)R_0^{-1}(P_0C' + N_0)' = 0 \tag{8-88}$$

对照式(8-88) 与前面提到的二次型调节器设计中的式(8-9)、式(8-10)就可以发现，二次型控制与二次型估计之间存在着对偶性质。如果对二次型调节器的求解作如下的代换，就可以得到最优滤波器的解。

$A \to A'$，$B \to C'$，$Q \to FQ_0F'$，$R \to R_0$，$N \to N_0$，

$K \to L'$，$P \to P_0$

8.6.2 MATLAB 实现

命令 lqe 用来求解最优滤波器的参数。其命令格式如下：

```
[L, P, E]= lqe(A, F, C, Q0 , R0 , N0)
```

式中，L 为 Kalman 滤波器增益矩阵；P 为对应 Riccati 方程的解，也是估计误差的协方差；E 为估计器的闭环特征值。N0 是可选项，它对应噪声相关项，即 $\mathrm{E}\{w(k)v'(k)\} = N_0$。

命令 estim 用来构成连续系统 Kalman 滤波器。其命令格式如下：

```
[Ae , Be , Ce , De]= estim(A, B, C, D, L)
```

由 Kalman 滤波器方程

$$\dot{\hat{x}} = (A - LC)\hat{x} + Ly$$

$$\begin{bmatrix} \hat{y} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} C \\ I \end{bmatrix} \hat{x}$$

可知，滤波器输出为传感器输出 y 的估计值 $\hat{y}$ 和状态 x 的估计值 $\hat{x}$，而滤波器的输入为传感器输出 y，且有如下对应关系：

$$A_e = A - LC \qquad B_e = L \qquad C_e = \begin{bmatrix} C \\ I \end{bmatrix}$$

上面的滤波器方程是将系统的输出全当作传感器输出(而需要估计)，而系统的输入只有随机噪声输入。实际上，系统的输出可以分成传感器输出和调节器输出。而系统的输入还会有其它的已知外部输入(如确定性的控制输入或外部给定输入)，这些输入并不用在滤波器设计阶段。因此，对这种情况，还有更一般形式的 estim 命令格式，如下所示：

[Ae, Be, Ce, De]=estim(A, B, C, D, L, sensors, known)

式中， sensors 是一个包含传感器输出序号的矢量； known 是包含外部输入序号的矢量。

【例 8-11 】给定单位反馈系统的受控对象和校正装置的传递函数如下：

$$G(s) = \frac{1}{s^2} \qquad H(s) = \frac{18(s+1)}{s+10}$$

求系统的最优滤波器，并进行仿真。

首先，用前几章介绍过的有关命令求取系统的状态方程。先将 $G(s)$和 $H(s)$表示为传递函数形式 numl 、 den1 、 num2 、 den2 ，再用传递函数到状态空间的转换命令 tf2ss 进行转换，并用 series 命令进行级联，最后用 cloop 构成单位反馈系统的状态方程(a_0，b_0， c_0， d_0)。然后在系统中加入有关的噪声。设系统的输入噪声为零均值、方差为 1 的高斯白噪声，传感器输出的量测噪声为零均值、方差为 0.01 的高斯白噪声。由于 randn 命令是用于产生零均值、方差为 1 的正态分布的随机数，因此，在加入到量测中时要乘以 0.1($\sqrt{0.01}$)。

下面给出的 MATLAB 程序 8-11 将先用 tf2ss 、 series 、 cloop 等命令构成系统，然后再加入有关噪声，并对系统进行仿真。量测噪声的加入是通过对仿真结果中的输出 y 加入方差为 0.01 的高斯白噪声来完成的。

```
% MATLAB PROGRAM 8-11
% Kalman-Bucy filter design
n1=[0 0 1];d1=[1 0 0];
[a1,b1,c1,d1]=tf2ss(n1,d1);
n2=[18 18];d2=[1 10];
[a2,b2,c2,d2]=tf2ss(n2,d2);
[a3,b3,c3,d3]=series(a1,b1,c1,d1,a2,b2,c2,d2);
[a0,b0,c0,d0]=cloop(a3,b3,c3,d3)
a0 =
     0   -18   162
     1     0     0
     0     1   -10
b0 =
     1
     0
     0
c0 =
     0    18  -162
d0 =
     0
dimf=size(b0);noinp=dimf(2);dimc=size(c0);
noout=dimc(1);
q0=1;w=sqrt(q0)*randn(100,noinp);
r0=0.01;nu=sqrt(r0)*randn(100,noout);
t=[0:0.1:(100-1)*0.1]';
[y,x]=lsim(a0,b0,c0,d0,w,t);
```

```
y=x*c0'+nu;
[L,P,E]=lqe(a0,b0,c0,q0,r0);L,P,E
L =
     9.0499
     1.0840
     0.0491
P =
     0.1089     0.0059     0.0001
     0.0059     0.0021     0.0002
     0.0001     0.0002     0.0000
E =
  -10.2766 + 8.6090i
  -10.2766 - 8.6090i
   -1.0065
```

现在再进一步地观察计算结果。滤波器的方程(无控制项存在)为

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

若量测噪声比较大(R_0 大)，而输入噪声较小(Q_0 小)，则滤波器取决于估计用的系统模型，且所选择的 L 是比较小的。由此得到的滤波器是一个较慢的滤波器。反之，若量测噪声比较小，而输入噪声较大的话，则滤波器将取决于量测值，且所选择的 L 是比较大的。由此得到的滤波器是有关较快的滤波器。在本例中，Q_0 比 R_0 大 10 倍，因此求得的滤波器应该是具有较大的 L 的快速滤波器。注意，L 中只有第一个元素比较大，而另两个元素则要小得多。为了说明此点，看一下系统方程本身

$$\begin{aligned}
\dot{x}_1 &= -18x_2 + 162x_3 + w \\
\dot{x}_2 &= x_1 \\
\dot{x}_3 &= x_2 - 10x_3 \\
y &= 10x_2 - 162x_3 + v
\end{aligned}$$

由于输入噪声直接加到 x_1 上，因此 x_1 上的噪声较大。但是 x_2 是

x_1 的积分，而 x_3 又是对 x_1 、 x_2 的滤波。很明显，当输入噪声到达 x_3 时，已经过了很强的滤波。这就是 $\boldsymbol{L}$ 中后两个元素要比第一个元素小得多的原因。

下面用 estim 命令来构成最优滤波器，并用 lsim 命令来仿真计算状态估计值，并将此次 lsim 得到的状态估计量 $\hat{\boldsymbol{x}}$ 与前面的 lsim 仿真计算得到的状态量 $\boldsymbol{x}$ 进行比较，并绘成图形。下面的 MATLAB 程序 8-12 用于构成滤波器、仿真及绘图，得到的图形见图 8-9 。图中(1)、(2)、(3)分别为 x_1 ， $\hat{x}_1$ ； x_2 ， $\hat{x}_2$ ； x_3 ， $\hat{x}_3$ 的对照，(4)为估计误差 $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \hat{\boldsymbol{x}}$ 的情况。由(4)可看出， x_3 的估计误差最多为 x_2 估计误差的 1/10 ，而且为 x_1 估计误差的 1/100(放在一起绘制时，可以看到 x_3 的完成曲线接近于 0)。

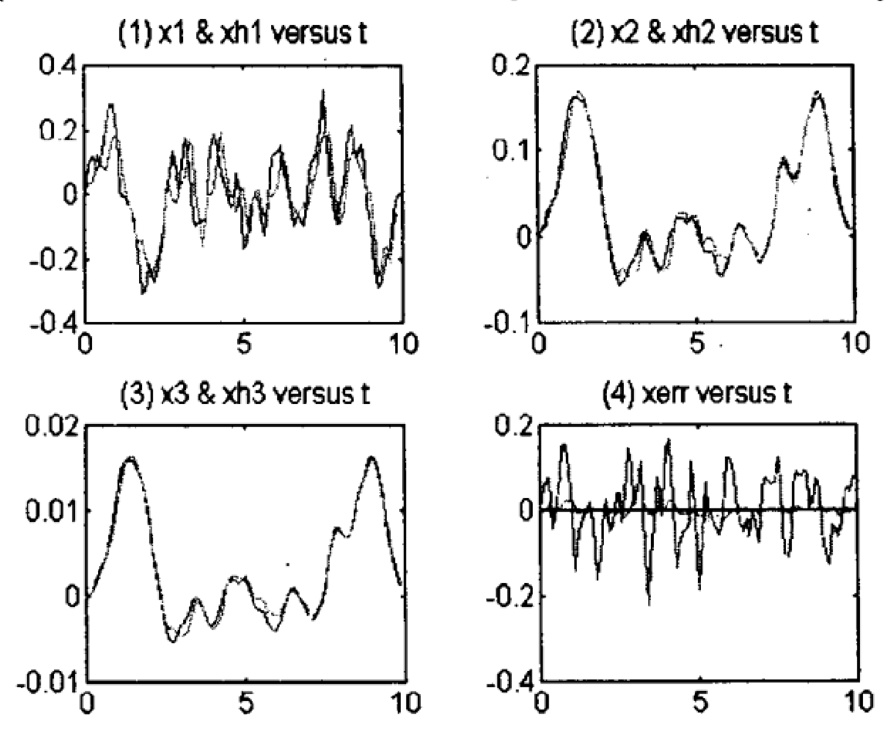


图 8-9　状态与其估计值的对照曲线

构成滤波器、仿真及绘图用 MATLAB 程序 8-12：

```
[a3, b3, c3, d3]=estim(a0, b0, c0, d0, L);
xh=lsim(a3, b3, c3, d3, y, t);
subplot(2, 2, 1);plot(t, x(:, 1), t, xh(:, 2))
title('(1) x1 & xh1 versus t')
```

```
subplot(2,2,2);plot(t,x(:,2),t,xh(:,3))
title('(2) x2 & xh2 versus t')
subplot(2,2,3);plot(t,x(:,3),t,xh(:,4))
title('(3) x3 & xh3 versus t')
xerr=x-xh(:,2:4);
subplot(2,2,4);plot(t,xerr)
title('(4) xerr versus t')
```

8.7 线性二次型高斯问题

在二次型最优控制中，若考虑到量测噪声和输入噪声对系统的影响，则称为线性二次型高斯问题，即 LQG 问题。最优控制就是由上节讨论的最优控制器得到的状态估计的线性函数。线性二次型调节器问题(LQR)是状态反馈问题，而线性二次型高斯问题则是输出反馈问题，它比 LQR 问题更具有实用性。

8.7.1 LQG 问题的求解

给定如下系统：

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}+\boldsymbol{F}\boldsymbol{w}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x}+\boldsymbol{v} \tag{8-89}$$

式中，$\boldsymbol{w}$ 为随机噪声干扰输入；$\boldsymbol{v}$ 为随机量测噪声。这两个噪声过程的统计特性与上节最优估计器中给出的相同，系统的性能指标为

$$J = \lim_{T\to\infty}\frac{1}{2T}\mathrm{E}\left\{\int_{-T}^{T}(\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x}+\boldsymbol{u}'\boldsymbol{R}\boldsymbol{u})\mathrm{d}t\right\} \tag{8-90}$$

由于系统的状态和控制都是随机的，因此性能指标函数也是随机的，即要进行最小化的性能指标是一个平均值。在与前几节讨论过的二次型调节器问题和最优估计器问题的假设条件相同的情况下，LQR 问题的解可以给出如下：

最优控制方程为

$$\boldsymbol{u} = -\boldsymbol{K}\hat{\boldsymbol{x}} \tag{8-91}$$

式中，最优反馈增益矩阵为

$$K = R^{-1}B'P \tag{8-92}$$

式中，P 为下面 Riccati 方程的解：

$$A'P+PA-PBR^{-1}B'P+Q = 0 \tag{8-93}$$

最优估计器方程为

$$\dot{\hat{x}} = A\hat{x}+Bu+L(y-C\hat{x}) \tag{8-94}$$

$$L = P_0C'R^{-1}$$

式中，P_0 为下列 Riccati 方程的解

$$A'P_0+P_0A-P_0C'R^{-1}CP_0+FQ_0F' = 0 \tag{8-95}$$

上述的 LQR 问题的求解公式与前几节的 LQR 问题和最优估计器问题的求解公式完全相同。 LQR 问题的求解满足分离定理，即在这种条件下的闭环最优控制的计算完全由分离的两部分来完成。第一部分是按照最优估计器的计算式(8-94)、式(8-95)来求得状态估计，第二部分是按确定性的 LQR 的计算式(8-92)、式(8-93)来求得最优反馈增益矩阵。最后，用 $\hat{x}(t)$代替 $x(t)$而得到最优控制 u 。

8.7.2 MATLAB 实现

命令 reg 用来构成 LQG 控制器。其命令格式如下：

[Ac, Bc, Cc, Dc]= reg(A, B, C, D, K, L)

式中， A 、 B 、 C 、 D 为受控制的连续系统参数； K 为由 lqr 命令得到的最优反馈增益矩阵； L 为由 lqe 命令得到的 Kalman 增益矩阵，假设系统的所有输入均为控制输入，而系统的所有输出均为传感器输出。

由下面的 LQG 控制器方程

$$\dot{\hat{x}} =[A-BK-LC+LDK]\hat{x}+Ly$$

$$\hat{u} = K\hat{x}$$

可知， LQG 是以控制反馈量 $\hat{u}$ 作为输出，而系统的传感器输出 y 作为输入，且有如下对应关系：

$$A_c = A-BK-LC+LDK \qquad B_c = L \quad C_c = K$$

LQG 控制器应该用负反馈来与原系统进行连接。

更一般形式的 reg 命令格式如下：

[Ac , Bc , Cc , Dc]= reg(A,B,C,D,K,L,sensors,known,controls)

式中， sensors 是一个包含传感器型输出序号的矢量； known 是包含附加外部输入序号的矢量； controls 为包含控制输入序号的矢量。

【例 8-12】 LQG 控制器设计。

设系统的状态方程如下：

$$\dot{x} = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -1.4 & 9.8 & -0.02 \end{bmatrix} x + \begin{bmatrix} 6.3 \\ 0 \\ 9.8 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} w$$

$$y = [0 \quad 0 \quad 1] x + v$$

二次型调节器性能指标为

$$J = \int_0^\infty (x' Q x + u' R u) \mathrm{d}t$$

式中

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R = 1$$

输入噪声和量测噪声的有关参数分别为 $Q_0 = 1$， $R_0 = 1$ 。

试设计 LQG 控制器，对得到的开环和闭环系统进行频域分析，并对系统进行仿真。

在 MATLAB 程序 8-13 中，先输入有关的系统参数。然后，由 lqr 命令和 lqe 命令分别求出最优反馈增益矩阵 $\boldsymbol{K}$ 和 Kalman 增益矩阵 $\boldsymbol{L}$ ，再由 reg 命令得到 LQG 控制器。为了进行频域分析，用 series 命令得到整个系统的开环状态方程($\boldsymbol{A}_{gk}$, $\boldsymbol{B}_{gk}$, $\boldsymbol{C}_{gk}$, $\boldsymbol{D}_{gk}$)，并用 feedback 命令得到闭环状态方程($\boldsymbol{A}_t$, $\boldsymbol{B}_t$, $\boldsymbol{C}_t$, $\boldsymbol{D}_t$)。通过用 bode 命令对开环和闭环状态方程的分析，得到了开环和闭环系统的对数幅相特性。程序中的 margin 命令用来得到系统的增

益裕量 GM 和相角裕量 PM，并得到系统的相位交界频率ω_{pc}和幅值交界频率ω_{gc}。同时还求得回差R_{sm}和闭环谐振峰值M_r。

在程序的最后，用 plot 命令给出了本系统的单位阶跃响应曲线，见图 8-10 。用 semilogx 命令分别绘出开环和闭环系统的伯德图，见图 8-12 和图 8-11 。

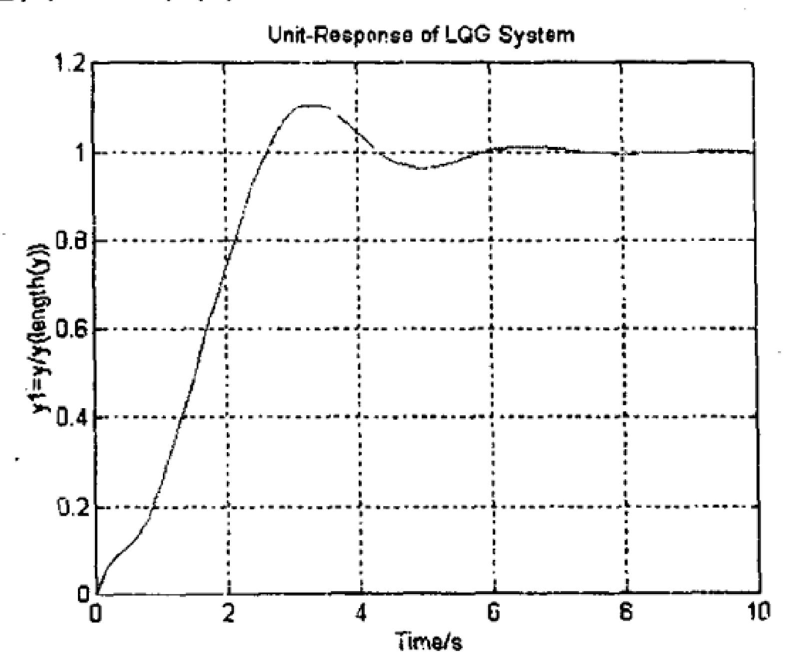


图 8-10　例 8-12 系统的单位阶跃响应

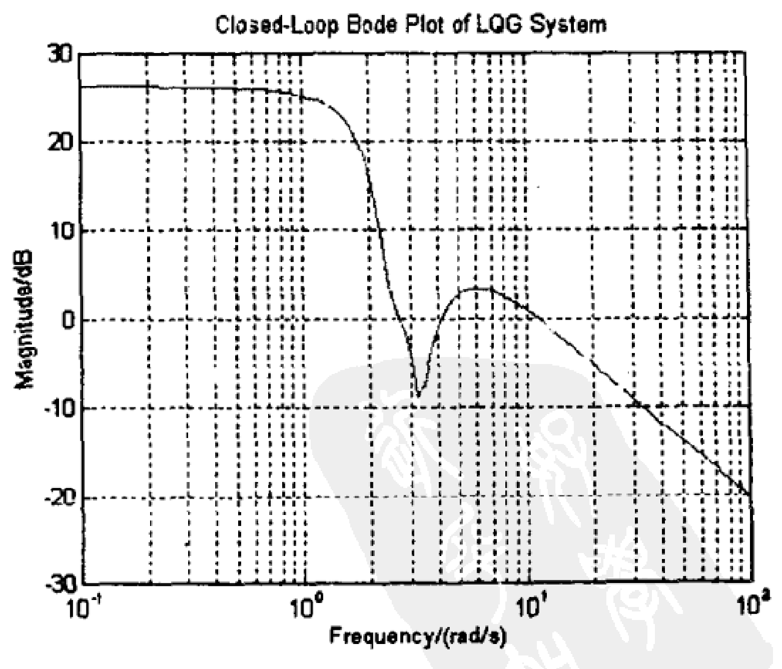


图 8-11　例 8-12 闭环系统伯德图

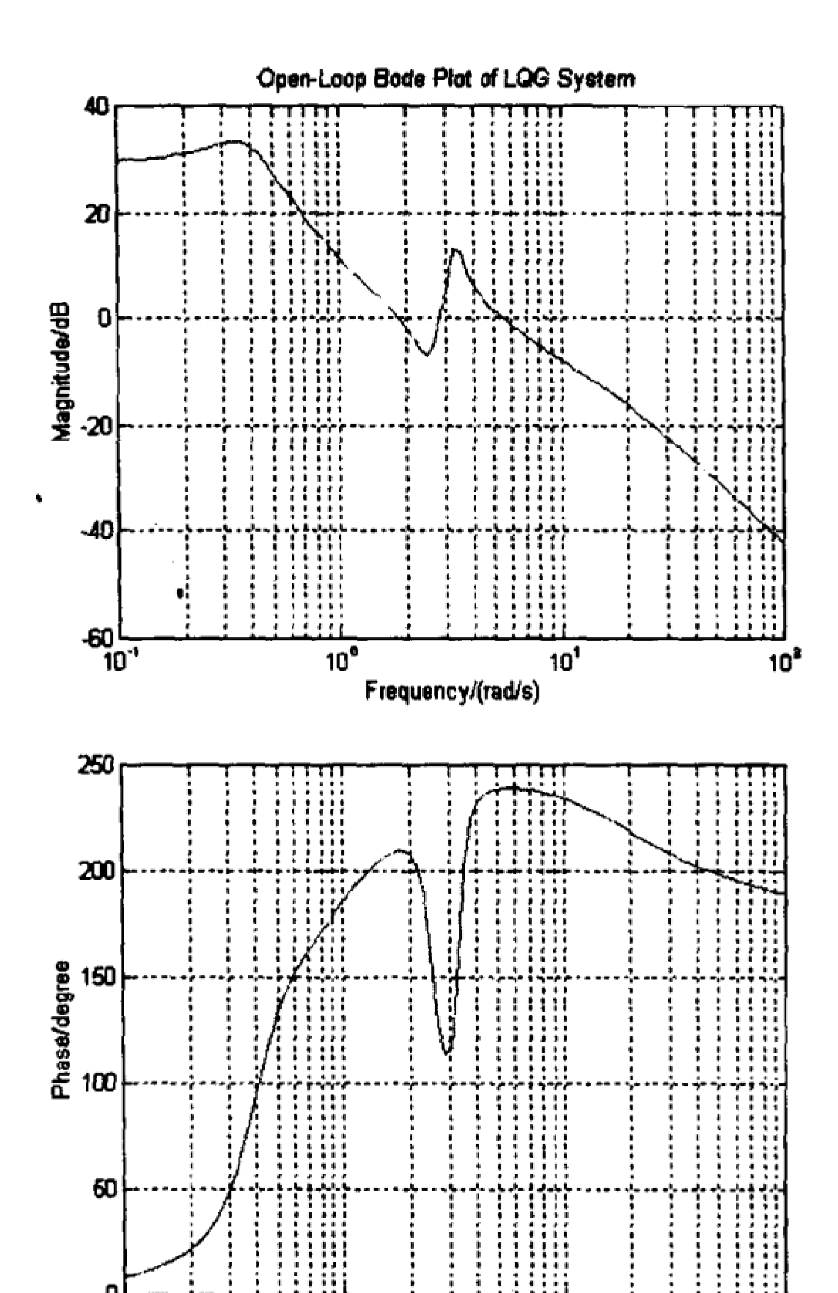


图 8-12 例 8-12 开环系统伯德图

```
% MATLAB PROGRAM 8-13
A=[-0.4 0 -0.01;1 0 0;-1.4 9.8 -0.02];B=[6.3;0;9.8];
C=[0 0 1];D=0;
F=[1;0;0];Q=[1 0 0;0 1 0;0 0 1];R=1;Q0=1;R0=1;
w=logspace(-1,2,100);
t=0:0.1:10;
```

```
K=lqr(A,B,Q,R);
L=lqe(A,F,C,Q0,R0);
[Ak,Bk,Ck,Dk]=reg(A,B,C,D,K,L);
[Agk,Bgk,Cgk,Dgk]=series(A,B,C,D,Ak,Bk,Ck,Dk);
[M,P]=bode(Agk,Bgk,Cgk,Dgk,1,w);
[At,Bt,Ct,Dt]=feedback(A,B,C,D,Ak,Bk,Ck,Dk);
[Mc,Pc]=bode(At,Bt,Ct,Dt,1,w);
[Ngk,Dgk]=ss2tf(Agk,Bgk,Cgk,Dgk,1);
f=freqs(Ngk,Dgk,w);
Ret=abs(ones(size(w))+f);
[GM,PM,Wpc,Wgc]=margin(M,P,w)
GM =
    2.0396
PM =
   29.5154
Wpc =
    2.3592
Wgc =
    1.8630
Rsm=min(Ret),Mr=max(Mc)
Rsm =
    0.4648
Mr =
   20.3521
y=step(At,Bt,Ct,Dt,1,t);y1=y/y(length(y));
plot(t,y1),grid
title('Unit-Response of LQG System')
xlabel('Time/s'), ylabel('y1=y/y(length(y))')
semilogx(w,20*log10(M)),grid
title('Open-Loop Bode Plot of LQG System')
```

```
xlabel('Frequency(rad/s)'),ylabel('Magnitude/dB')
semilogx(w,P),grid
xlabel('Frequency(rad/s)'),ylabel('Phase/degree')
semilogx(w,20*log10(Mc)),grid
title('Closed-Loop Bode Plot of LQG System')
xlabel('Frequency(rad/s)');ylabel('Magnitude/dB')
```

习　题

8-1　考虑如下双积分系统：

$$G(s)=\frac{1}{s^2}$$

或

$$\dot{x}=\begin{bmatrix}0 & 1\\0 & 0\end{bmatrix}x+\begin{bmatrix}0\\1\end{bmatrix}u \quad y=\begin{bmatrix}1 & 0\end{bmatrix}x$$

令$Q=\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$、$R=1$，设计出满足这一线性二次型指标的最优控制器。

8-2　已知受控系统的状态方程为

$$\dot{x}=\begin{bmatrix}-0.4 & 0 & -0.01\\1 & 0 & 0\\-1.4 & 9.8 & -0.02\end{bmatrix}x+\begin{bmatrix}6.3\\0\\9.8\end{bmatrix}u$$

$$y=\begin{bmatrix}0 & 0 & 1\end{bmatrix}x$$

设计该系统的线性二次型调节器。设二次型性能指标中的$Q=qI$、$R=r$。观察q、r变化时对系统性能的影响(设分为$q=1$、10、100三种情况，$r=1$不变；及$r=1$、10、100三种情况，$q=1$不变)。绘出上述几种情况时的单位阶跃响应曲线。

8-3　已知离散系统的状态方程为

$$x(k+1)=Ax(k)+Bu(k)$$

式中，$A=\begin{bmatrix}0 & 1\\-1 & 1\end{bmatrix}$，$B=\begin{bmatrix}0\\1\end{bmatrix}$，初始状态分别为$x(0)=\begin{bmatrix}1\\1\end{bmatrix}$，$x(10)=\begin{bmatrix}0\\0\end{bmatrix}$。求最优控制$K(i)$，使得性能指标

$$J=\frac{1}{2}\sum_{i=0}^{9}[x^2(i)+u^2(i)]$$

达到最小值。

8-4 给定二阶系统

$$x(k+1)=Ax(k)+Bu(k)$$

式中，$A=\begin{bmatrix}0.5 & 1\\ -1 & 1\end{bmatrix}$，$B=\begin{bmatrix}0\\ 1\end{bmatrix}$，初始状态$\begin{bmatrix}x_1(0)\\ x_2(0)\end{bmatrix}=\begin{bmatrix}10\\ -10\end{bmatrix}$。分别求出使得系统在 3 个采样周期和 4 个采样周期内将初始状态转移到原点，且使如下性能指标

$$J=\frac{1}{2}\sum_{k=0}^{n}u^2(k)\qquad (n=3,\ 4)$$

达到极小的控制序列。

8-5 考虑习题 8-1 所示系统，现再加入输入噪声和量测噪声。受控对象的状态方程如下：

$$\dot{x}=Ax+Bu+w$$
$$y=Cx+v$$

设计本系统的 Kalman 滤波器，设有关的噪声过程的参数如下：

$$Q_0=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\qquad R_0=1$$

求出此 LQR 控制器的传递函数，并由此求出开环系统伯德图及相角和增益裕量。

第9章　系统设计实例

9.1　概述

在前几章中，较详细地介绍了控制系统设计中常用的根轨迹法、超前滞后校正法、状态反馈法及有关数字调节器的设计方法。应用MATLAB语言，特别是控制工具箱的有关命令设计调节器系统，简化了设计过程及步骤。通过改变系统参数和补偿器参数，可以方便地得到不同的频域特性和时域特性，根据性能指标，选取一组最佳参数，从而完成系统的设计和仿真。不同的系统和不同的性能指标，反映在MATLAB设计程序上，仅仅改变MATLAB程序中的系统矩阵参数及期望极点的位置。因此，进行系统设计和仿真是简单易行的。

本章将对有关连续系统和离散系统常见的设计方法进行介绍，并对有关问题进行讨论。由于连续系统的应用举例在前几章已经很多，因此这里着重于离散系统的分析举例。

9.2　连续系统设计实例

在这一节中，将详细讨论典型Ⅰ型伺服系统的设计问题。假设系统为

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}u\\ y &= \boldsymbol{C}\boldsymbol{x}\end{aligned} \tag{9-1}$$

式中，$\boldsymbol{x}$为n维矢量；u为输入信号；y为输出信号；$\boldsymbol{A}$为$n\times n$维常数矩阵；$\boldsymbol{B}$为$n\times 1$维常数矩阵；$\boldsymbol{C}$为$1\times n$维常数矩阵。

图9-1 给出了典型Ⅰ型伺服系统的结构图，假定$y=x_1$。在该系统中，采用下列状态反馈控制方案：

$$\begin{aligned}u &= -[0\ \ k_2\ \ k_3\ \ \cdots\ \ k_n\ ]\boldsymbol{x}+k_1(r-x_1)\\ &= -\boldsymbol{K}\boldsymbol{x}+k_1 r\end{aligned} \tag{9-2}$$

式中　$\boldsymbol{K}=[k_1\quad k_2\quad k_3\quad \cdots\quad k_n\ ]$

假设输入r是单位阶跃输入，并在$t\geqslant 0$时加在系统输入端。

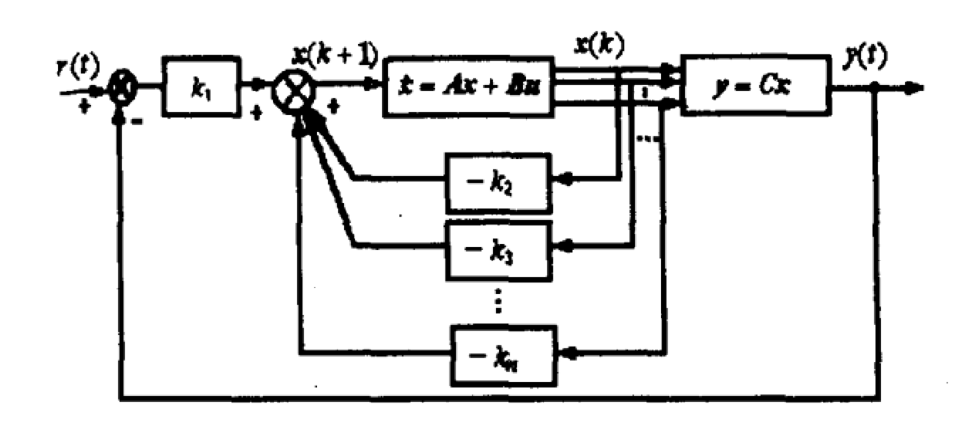


图9-1　系统结构框图

由方程式(9-1)和式(9-2)所表示的系统可改写为

$$\begin{aligned}\dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u \\ &= (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x} + \boldsymbol{B}k_1 r\end{aligned} \tag{9-3}$$

设计状态反馈$\boldsymbol{K}$，以得到期望的极点。在稳态时有

$$\dot{\boldsymbol{x}}(\infty) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}(\infty) + \boldsymbol{B}k_1 r(\infty) \tag{9-4}$$

r是阶跃输入，因此$r(\infty)=r$。由于方程式(9-3)减去式(9-4)有

$$\dot{\boldsymbol{x}}(t) - \dot{\boldsymbol{x}}(\infty) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})(\boldsymbol{x}(t) - \boldsymbol{x}(\infty)) \tag{9-5}$$

令$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{x}(\infty)$，则方程式(9-5)为

$$\dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{e} \tag{9-6}$$

方程式(9-6)表示了系统的误差特性。

如果系统是状态完全能控的，则可以实现任意极点配置。下面举一个例子。

【例9-1】考虑下述系统：

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u$$
$$y = \boldsymbol{C}\boldsymbol{x}$$

式中

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

参照图9-1，控制信号u为

$$u = -\boldsymbol{K}\boldsymbol{x} + k_1 r$$

式中，　$\boldsymbol{K} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$

系统的期望极点是$-2 \pm \mathrm{j}2\sqrt{3}$，$-10$。

应用下列控制工具箱中的place命令和ctrl命令可以确定状态

反馈增益矩阵 $\boldsymbol{K}$：

```
M=ctrb(A,B,C,D);
rank(M)
ans=
    3
```

能控矩阵的秩是3，因此系统是状态完全能控的。

```
p=[-10,-2+2*sqrt(3)*j,-2-2*sqrt(3)*j];
K=place(A,B,p)
K=
  160.0000     54.0000     11.0000
```

下面分析所设计系统的阶跃响应特性。所设计系统是

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \boldsymbol{BK})\boldsymbol{x} + \boldsymbol{B}k_1 r$$

$$y = \boldsymbol{Cx}$$

MATLAB程序9-1是求取单位阶跃响应的程序清单。

```
%  MATLAB PROGRAM 9-1
% Pole placement
A=[0 1 0;0 0 1;0 -2 -3];
B=[0;0;1];
C=[1 0 0];D=0;
w=[0.01 10];
t=[0:0.05:10];
M=ctrb(A,B);
rank(M)
P=[-10 -2+2*sqrt(3)*j -2-2*sqrt(3)*j];
[olz olp olg]=ss2zp(A,B,C,D);
K=place(A,B,p)
bode(A,B,K,0);
[gm pm wpc wgc]=margin(A,B,K,0);
figure
rlocus(A,B,K,0)
axis([-4 1 -5 5])
At=A-B*K;Bt=B;
Ct=C;Dt=D;
[zT pT gainT]=ss2zp(At,Bt,Ct,Dt);
```

```
dcg=dcgain(At,Bt,Ct,Dt);
figure
[yc x t]=step(At,Bt,Ct,Dt,1,t);
yc1=yc/dcg;
plot(t,yc1)
xlabel('t  / s')
ylabel('y  output')
grid;
axis([0 3 0 1.2])
figure
x1=[1 0 0]*x';
plot(t,x1)
xlabel('t  / s')
ylabel('x1  state')
grid;
axis([0 3 0 0.01])
figure
x2=[0 1 0]*x';
plot(t,x2)
xlabel('t  / s')
ylabel('x2  state')
grid;
axis([0 3 -0.01 0.02])
figure
x3=[0 0 1]*x';
plot(t,x3)
xlabel('t  / s')
ylabel('x3  state')
grid;
axis([0 3 -0.04 0.06])
end
```

图9-2给出了运行上述程序后得到的系统单位阶跃响应曲线，由曲线可见，系统的过渡过程时间大约为2.5s，超调量为15%，大大地改善了系统的性能指标。

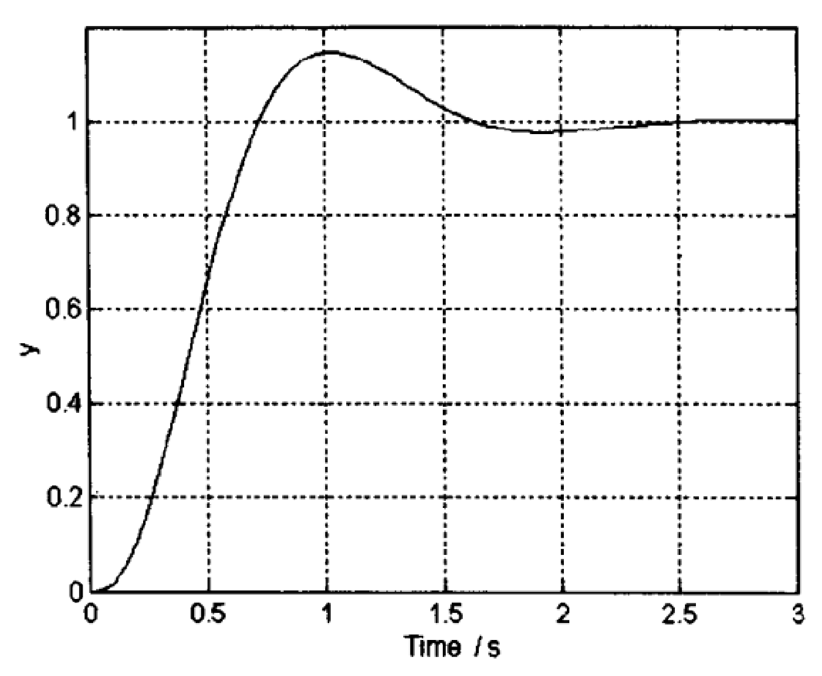


图9-2　系统的单位阶跃响应

图9-3给出了系统状态变量的响应曲线。

下面讨论一下状态观测器的设计，前面第5章已经对观测器的设计进行了分析和举例。一般来说，观测器增益矩阵$\boldsymbol{K}_e$取决于期望的特征方程

$$(s-\mu_1)(s-\mu_2)\cdots(s-\mu_n)=0$$

式中，μ_1、μ_2、…、μ_n的选取不是唯一的。期望的特征值和特征方程必须保证状态观测器的响应比闭环系统的响应快2～5倍。

假如有不同的期望的特征方程，决定着不同的观测器增益矩阵$\boldsymbol{K}_e$，可以对所设计的$\boldsymbol{K}_e$进行仿真实验，观测系统性能，然后求取最佳矩阵$\boldsymbol{K}_e$。在许多实际情况下，必须综合地考虑系统的快速性和对干扰及噪声的敏感度。

【例9-2】考虑如下的调节器系统状态方程：

$$\dot{x}=Ax+Bu$$
$$y=Cx$$

式中

$$A=\begin{bmatrix}0 & 1\\ 20.6 & 0\end{bmatrix}\qquad B=\begin{bmatrix}0\\ 1\end{bmatrix}\qquad C=\begin{bmatrix}1 & 0\end{bmatrix}$$

采用极点配置方法设计系统，系统的期望极点为μ_1=-1.8+j2.4

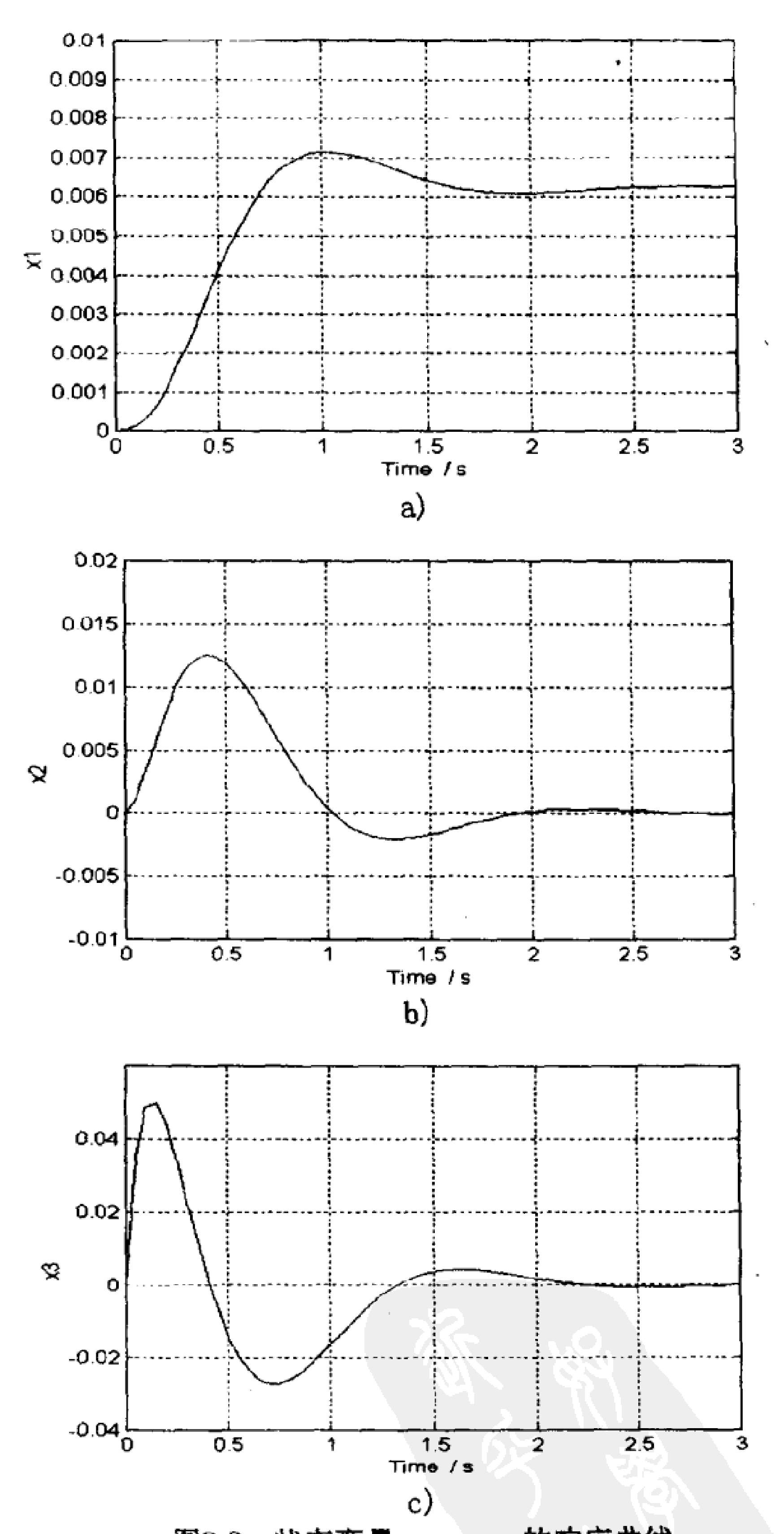


图9-3　状态变量x_1、x_2、x_3的响应曲线

a）状态变量x_1的响应曲线　b）状态变量x_2的响应曲线　c）状态变量x_3的响应曲线

和 μ_2=-1.8-j2.4，则系统的状态反馈增益矩阵 $\boldsymbol{K}$ 可以用MATLAB程序9-2求得。

```
% MATLAB PROGRAM 9-2
A=[0 1;20.6 0];B=[0;1];C=[1 0];D=0;
P=[-1.8+j*2.4 -1.8-j*2.4];
K=place(A,B,P)
end
K =
    29.6000      3.6000
```

用状态反馈 $\boldsymbol{K}$，控制信号 u 为

$$u=-\boldsymbol{K}\boldsymbol{x}=-\begin{bmatrix}29.6 & 3.6\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}$$

采用观测的状态 $\tilde{\boldsymbol{x}}$ 代替实际状态 $\boldsymbol{x}$，则有

$$u=-\boldsymbol{K}\tilde{\boldsymbol{x}}$$

选取观测器增益矩阵 $\boldsymbol{K}_e$ 的特征值为

$$\mu_1=\mu_2=-8$$

则由MATLAB命令可以求出增益矩阵 $\boldsymbol{K}_e$。

```
op=[-8 -8];
Ke=place(A',C',op);Ke=Ke'
Ke =
     16
     84.6
```

根据设计参数，可以画出带状态反馈 $\boldsymbol{K}$ 和观测器 $\boldsymbol{K}_e$ 的系统框图，见图9-4。

MATLAB程序9-3是设计系统和求取系统动态响应的程序清单。

```
% MATLAB PROGRAM 9-3
% Observer  design
A=[0 1;20.6 0];B=[0;1];C=[1 0];D=0;
w=[0.01 10];t=[0:0.05:10];
P=[-1.8+j*2.4 -1.8-j*2.4];op=[-7.9998 -8.0002];
x0=[1 2];xh0=[-1 -2];
K=place(A,B,P)
L=place(A',C',op);L=L'
```

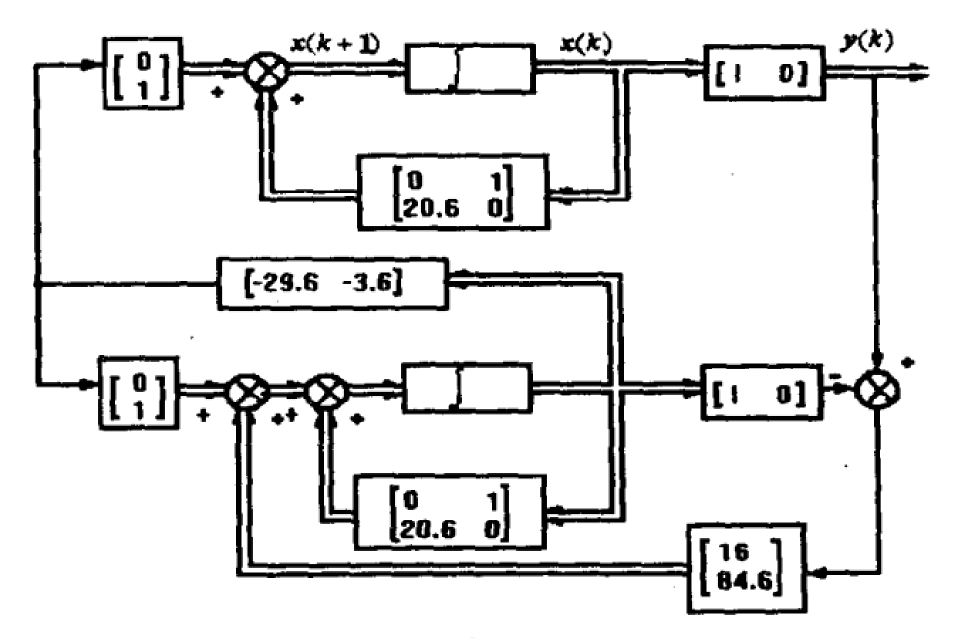


图9-4 系统框图

```
Ak=A-B*K-L*C;Bk=L;Ck=K;
[tmp1 tmp2]=size(D);
Dk=zeros(tmp1,tmp2);
[At Bt Ct Dt]=feedback(A,B,C,D,Ak,Bk,Ck,Dk);
dcg=dcgain(At,Bt,Ct,Dt);
yc=step(At,Bt,Ct,Dt,1,t);yc=yc/dcg;
At1=A-B*K;Bt1=B;Ct1=C;Dt1=D;
dcg=dcgain(At1,Bt1,Ct1,Dt1);
yc1=step(At1,Bt1,Ct1,Dt1,1,t);yc1=yc1/dcg;
figure;plot(t,yc);
xlabel('t  / s');ylabel('yc')
axis([0 3.5 0 1.2]);grid
figure;plot(t,yc,'r',t,yc1,'g');
xlabel('t  / s');ylabel('yc  yc1')
axis([0 3.5 0 1.2]);grid
[ata bta cta dta]=augstate(At,Bt,Ct,Dt);
t2=[0:0.02:4]';
x_xh=initial(ata,bta,cta,dta,[x0 xh0],t2);
x1=x_xh(:,2);xh1=x_xh(:,4);
x2=x_xh(:,3);xh2=x_xh(:,5);
figure;plot(t2,x1,t2,xh1);
```

```
xlabel('t   / s');ylabel('x1   xh1')
axis([0 3 -1 2]);grid
figure;plot(t2,x2,t2,xh2);
xlabel('t   / s');ylabel('x2   xh2')
axis([0 3 -4 6]);grid
end
```

图9-5是所设计系统的阶跃响应曲线。由图中可以看出，系统的过渡过程时间是3s，超调量是10%。

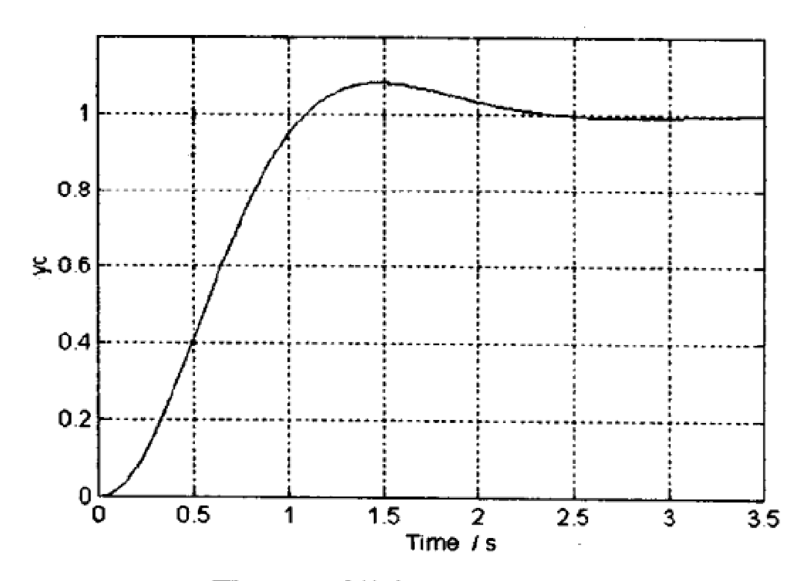


图9-5　系统的阶跃响应曲线

图9-6给出了加状态反馈$\boldsymbol{K}$的系统响应曲线，和加入观测器$\boldsymbol{K}_e$后系统的响应曲线。

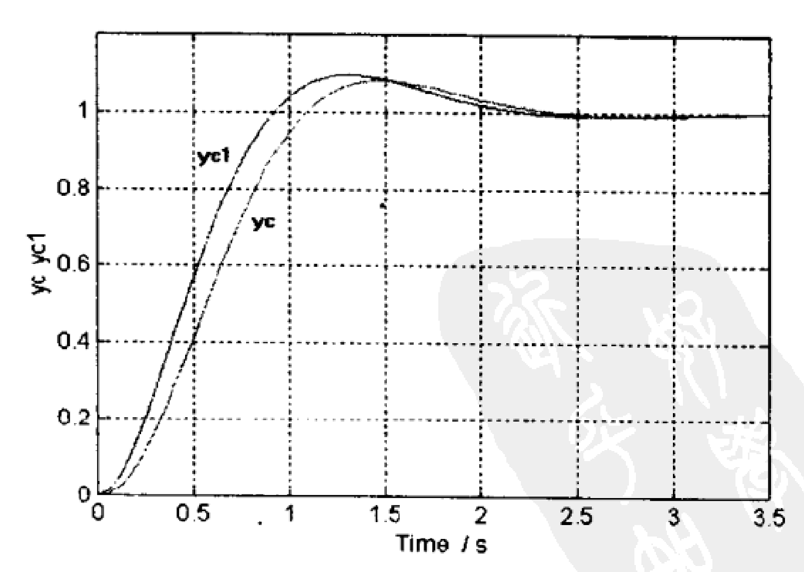


图9-6　加状态反馈矩阵$\boldsymbol{K}$和加观测器矩阵$\boldsymbol{K}_e$的响应曲线

图9-7给出了系统状态和观测状态的动态响应曲线。可以看出，图9-6和图9-7动态响应是对应的。当估测状态和实际状态一致后，则系统的响应也趋于相同，且系统的初始值不影响响系统的稳定性。

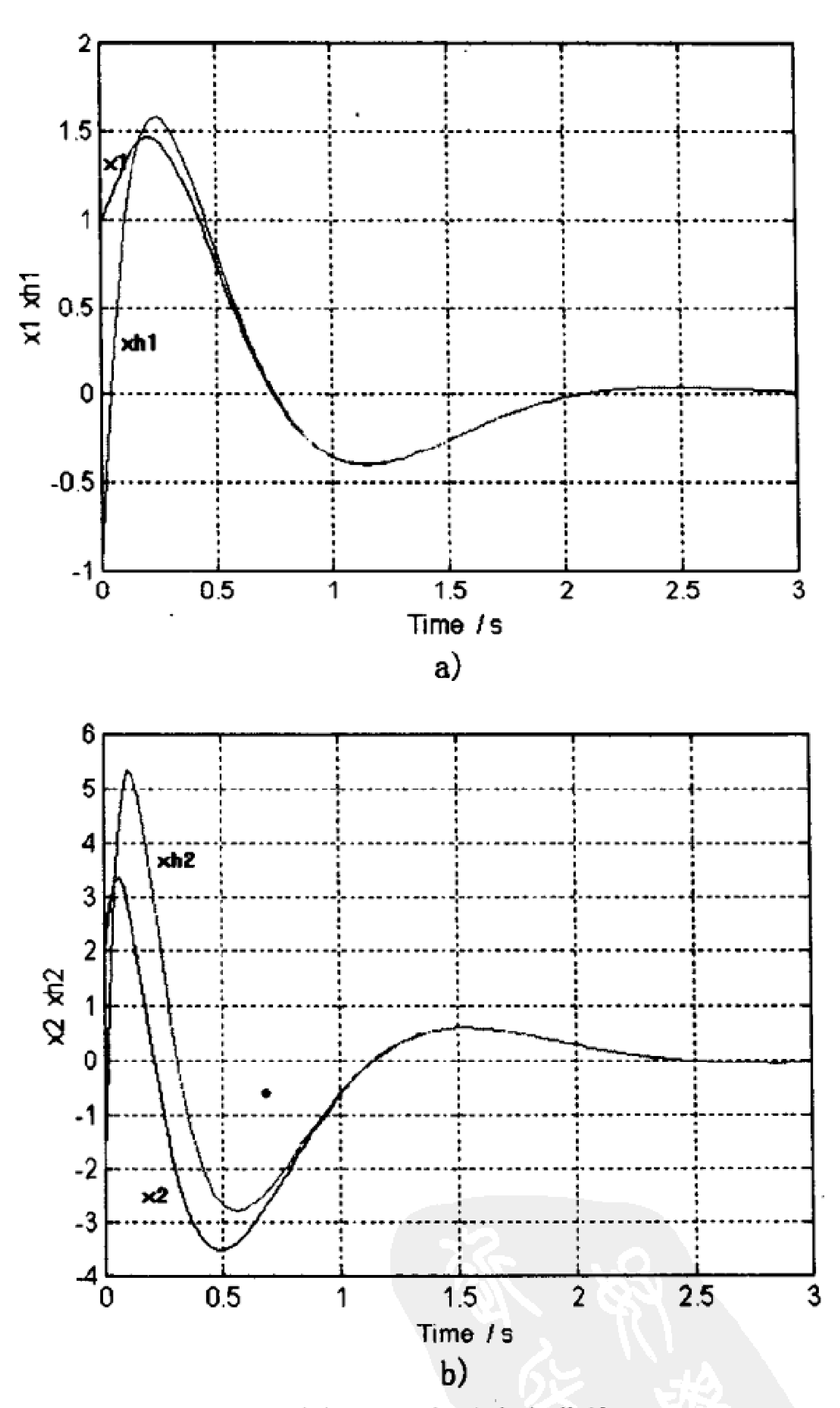


图9-7 阶跃响应曲线

a）系统估测状态xh_1和实际状态x_1的阶跃响应曲线

b）系统估测状态xh_2和实际状态x_2的阶跃响应曲线

9.3 离散系统设计实例

9.3.1 离散系统根轨迹的设计实例

离散系统在z平面根轨迹的图形与连续系统在s平面的图形基本上是相同的，唯一的差别是两个平面上图形的稳定区域的解释不同，在s平面上，闭环系统的极点在右半平面是不稳定的，而在z平面上，闭环极点在单位圆外是不稳定的。

【例9-3】假设系统的开环传递函数为

$$G_D(z)G(z)=\frac{0.0176K(z+0.8760)}{(z-0.2543)(z-1)}$$

为绘制该系统的根轨迹，令K'=0.0176K，K'从0变化到非常大的数，因此上式被写为

$$G_D(z)G(z)=\frac{K'(z+0.8760)}{(z-0.2543)(z-1)}=\frac{K'(z+0.8760)}{z^2-1.2543z+0.2543}$$

于是有

```
num=[0 1 0.8760];
den=[1 -1.2543 0.2543];
```

MATLAB程序9-4是求取系统根轨迹的程序清单。

```
% MATLAB PROGRAM 9-4
num=[0 1 0.8760];
den=[1 -1.2543 0.2543];
rlocus(num,den);
grid
v=[-4 2 -3 3];
axis(v);
end
```

其根轨迹见图9-8。

【例9-4】系统结构框图见9-9，绘制其根轨迹图。

上述开环系统传递函数为

$$G_D(z)G(z)=\frac{(1.4-1.4z^{-1}+0.2z^{-2})(0.3679z^{-1}+0.2642z^{-2})}{(1-z^{-1})^2(1-0.3679z^{-1})}$$

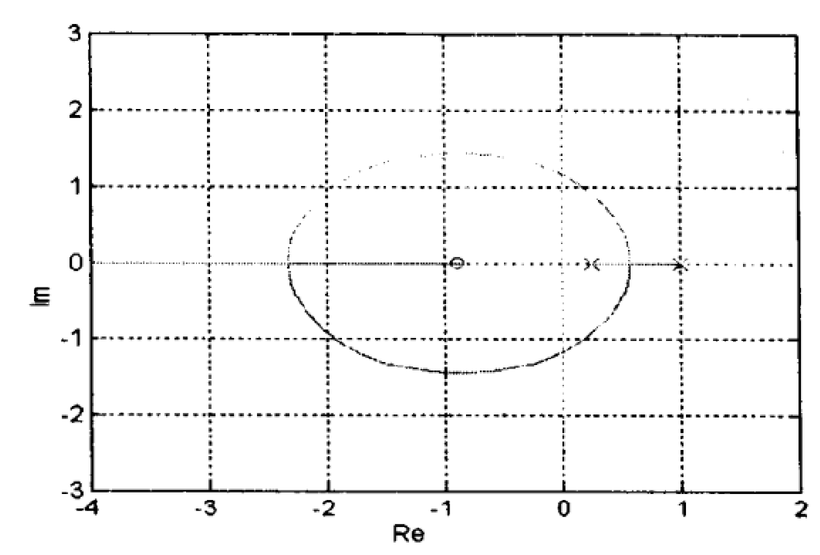


图9-8　$K(z+0.8760)/[(z-0.2543)(z-1)]$ 的根轨迹

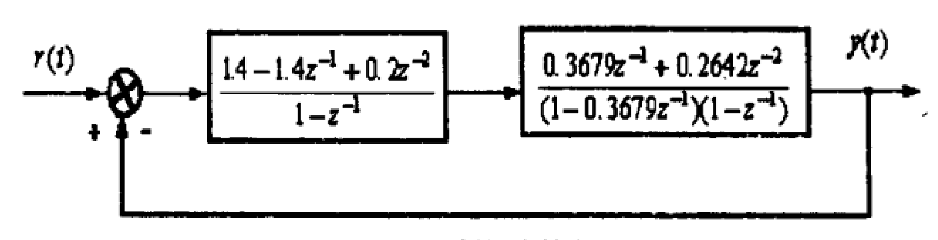


图9-9　系统结构框图

$$G_D(z)G(z)=\frac{0.5151z^3+0.1452z^2-0.2963z+0.65284}{z^4-2.3679z^3+1.7358z^2-0.3679z}$$

MATLAB程序9-5是求取系统根轨迹的程序清单。

```
% MATLAB PROGRAM 9-5
num=[0.5151 -0.1452 -0.2963 0.05284];
den=[1 -2.3679 1.7358 -0.3679 0];
rlocus(num,den);
grid
v=[-3 2 -2 2];
axis(v);
end
```

其根轨迹图见图9-10。

【例9-5】 求取图9-11所示系统的根轨迹。

为了更方便地判断系统的稳定性，在程序内都加入了单位圆，如图9-12中虚线所示。

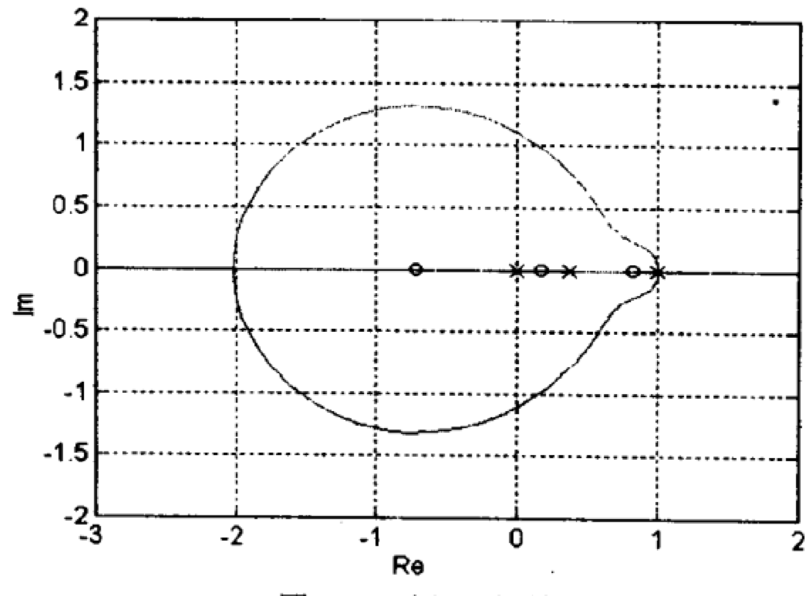


图9-10　例9-4根轨迹图

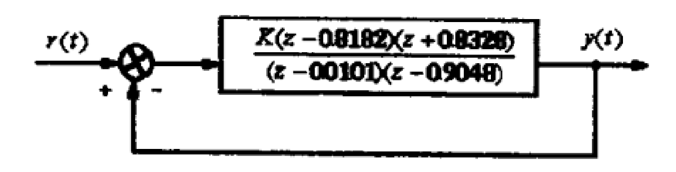


图9-11　系统结构框图

MATLAB程序9-6是求取系统根轨迹的程序清单。

```
% MATLAB program 9-6
num=[0 1 0.0146 -0.6814];
den=[1 -1.9149 0.9240 -0.0091];
rlocus(num,den);
grid
hold
p=0:0.07:2*pi;
x=sin(p);
y=cos(p);
plot(x,y,'.')
hold
xlabel('real axis')
ylabel('imag axis')
v=[-4 2 -2 2];
axis(v)
```

```
end
```

其根轨迹图见图9-12。

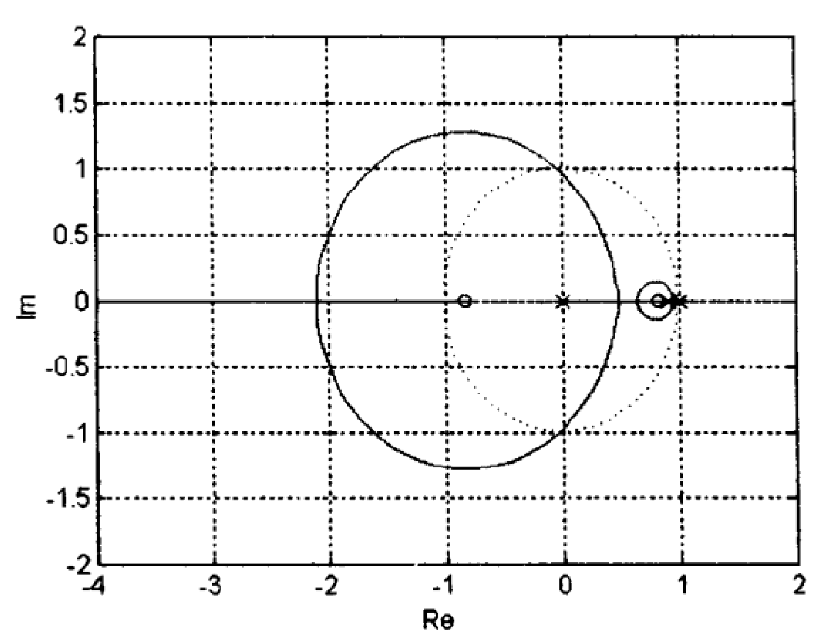


图9-12　例9-5根轨迹图

【例9-6】 讨论图9-13所示的非调谐控制系统，其开环传递函数如下：

$$G_D(z)G(z)=\frac{0.5820(z+0.7181)}{(z+0.4180)(z-1)}=\frac{0.5820(z+0.7181)}{z^2-0.5820z-0.4180}$$

$R(z)$ → $\frac{1.5820(z-0.3679)}{z+0.4180}$ → $\frac{0.3679(z+0.7181)}{(z-1)(z-0.3679)}$ → $C(z)$（单位负反馈，+、-）

图9-13　非调谐系统的结构框图

注意，非调谐系统的所有闭环极点都在原点附近，该闭环系统传递函数如下：

$$\frac{C(z)}{R(z)}=\frac{0.5820(z+0.7181)}{(z+0.4180)(z-1)+0.5820(z+0.7181)}$$

$$=\frac{0.5820(z+0.7181)}{z^2}$$

其两个闭环极点均在原点上，利用MATLAB程序9-7可以得到其根轨迹图。

```
% MATLAB PROGRAM 9-7
```

```
num=[0 1 0.7181];
den=[1 -0.5820 -0.4180];
r=rlocus(num,den);
plot(r,'-')
xlabel('real axis')
ylabel('imag axis')
v=[-2 0.5 -1 1];
axis(v)
end
```

其根轨迹图见图9-14。

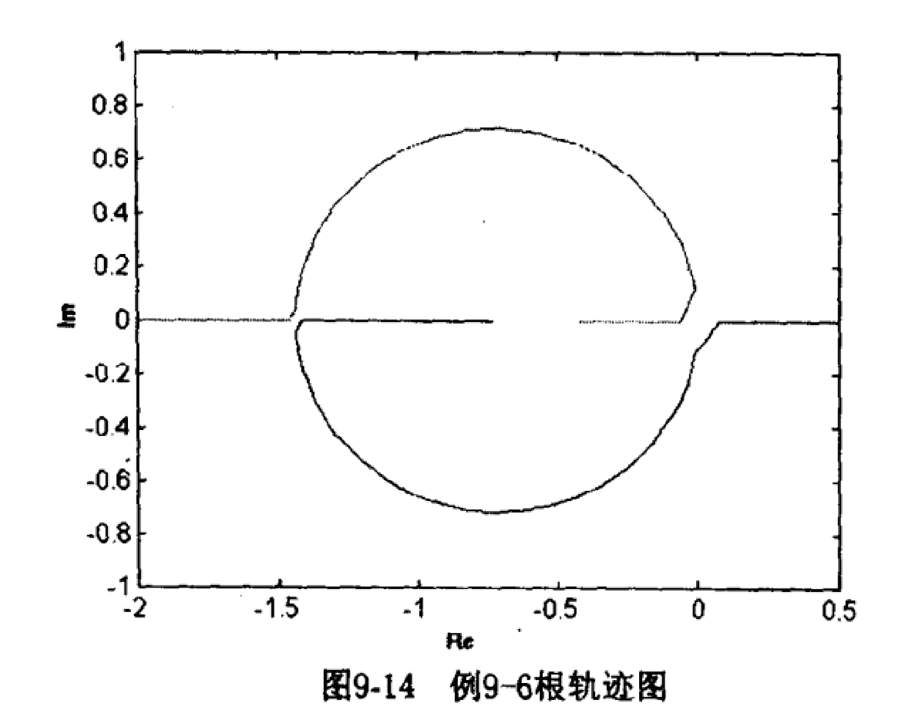


图9-14　例9-6根轨迹图

仔细地检查会发现，闭环系统的根轨迹并没有通过原点，错误的根轨迹发生在原点和(-1.4,0)点附近，这主要是由于该两点的步距较大造成的，即K值变化过大造成的。

可以采用在该两点附近取较小的K值变化来校正上述错误，MATLAB程序9-8是设计程序清单。

```
% MATLAB PROGRAM 9-8
num=[0 1 0.7181];
den=[1 -0.5820 -0.4180];
k1=0:0.1:0.5;
k2=0.5:0.0011:0.6;
```

```
k3=0.6:0.1:3.4;
k4=3.4:0.002:3.6;
k5=3.6:1:30.6;
k=[k1 k2 k3 k4 k5];
r=rlocus(num,den,k);
plot(r,'-'),grid
xlabel('real axis')
ylabel('imag axis')
v=[-2 0.5 -1 1];
axis(v)
end
```

其校正后的根轨迹见图9-15。

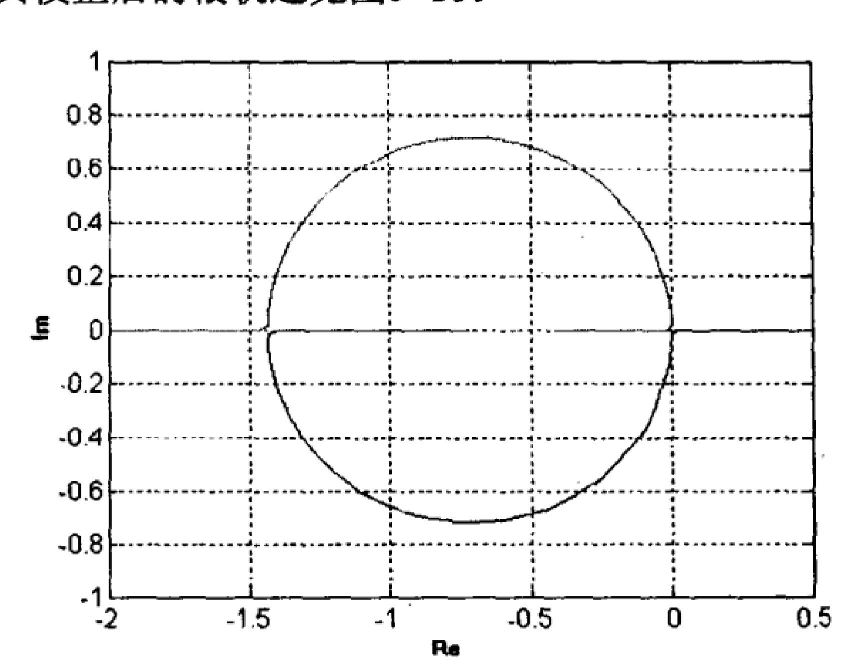


图9-15　校正后的根轨迹图

9.3.2　离散系统动态响应分析设计实例

离散系统动态响应的MATLAB命令与连续系统时不同，对于离散系统，常使用的命令格式是

```
y = filter(num,den,x)
```

其中，x是输入，y是输出。

在绘制系统动态响应曲线前，首先讨论一下离散系统输入函数的形式。

脉冲函数：脉冲函数的定义为

$$u(0)=1$$

$$u(k)=0 \qquad (k=1,2,3,4,\cdots)$$

其输入为

$$u(0)=1$$

$$u(k)=0 \qquad (k=1,2,3,\cdots,60)$$

则在MATLAB程序中可以写成

```
u =[1  zeros(1,60)]
```

例如，脉冲幅值为8的输入信号,其形式为

$$u(0)=8$$

$$u(k)=0, \qquad (k=1,2,3,\cdots,40)$$

在MATLAB程序内可以写为

```
u=[8,zeros(1,40)]
```

阶跃输入：其阶跃输入的形式为

$$u(k)=1 \qquad (k=0,1,2,\cdots,100)$$

则在MATLAB程序中可以写为

```
u=[1,ones(1,100)]
```

或 `u=ones(1,101)`

同样，幅值为5的阶跃输入

$$u(k)=5 \qquad (k=0,1,2,3,\cdots,50)$$

在MATLAB程序中可以写为

```
u=5*ones(1,51)
```

斜坡输入：单位斜坡输入为

$$u=t \qquad (t\geqslant 0)$$

在离散系统中$t=kT$，T是采样周期，斜坡输入为

$$u(k)=kT \qquad (k=0,1,2,\cdots,50)$$

则在MATLAB程序中有如下形式：

```
k=0:50; u=(k*T);
```

若T=0.2s，k=50，则有

```
k=0:50; u=(0.2*k);
```

加速度输入信号：加速度输入的一般形式为

$$u(k)=\frac{1}{2}(kT)^2 \qquad (k=0,\ 1,\ 2,\ \cdots)$$

因此，在MATLAB程序中，当取10步，T=0.2s时，可以写成如下形式：

```
k=0:10;u=[0.5*(0.2*k).^2];
```

假若数字控制系统的闭环传递函数为

$$\frac{C(z)}{R(z)}=\frac{\mathrm{num}(z)}{\mathrm{den}(z)}$$

其输入信号为u，则

y=filter(num,den,u)

从而得到y的一组数据，即为对输入的响应，下面给出几个例题。

【例9-7】 求取下列传递函数的脉冲响应：

$$\frac{Y(z)}{X(z)}=G(z)=\frac{0.01409z^3+0.02818z^2+0.01409z}{z^3-2.7624z^2+2.5811z-0.8187}$$

脉冲输入形式为

$x=\{1,\mathrm{zeros}(1,40)\}$

MATLAB程序9-9是求取系统脉冲响应的程序清单：

```
% MATLAB PROGRAM 9-9
num=[0.01409 0.02818 0.01409 0];
den=[1 -2.7624 2.5811 -0.8187];
x=[1,zeros(1,50)];
y=filter(num,den,x);
plot(y,'g-')
xlabel('NO. of sample');ylabel('y')
axis([0 70 0 1.4]);grid
end
```

其脉冲响应曲线见图9-16。

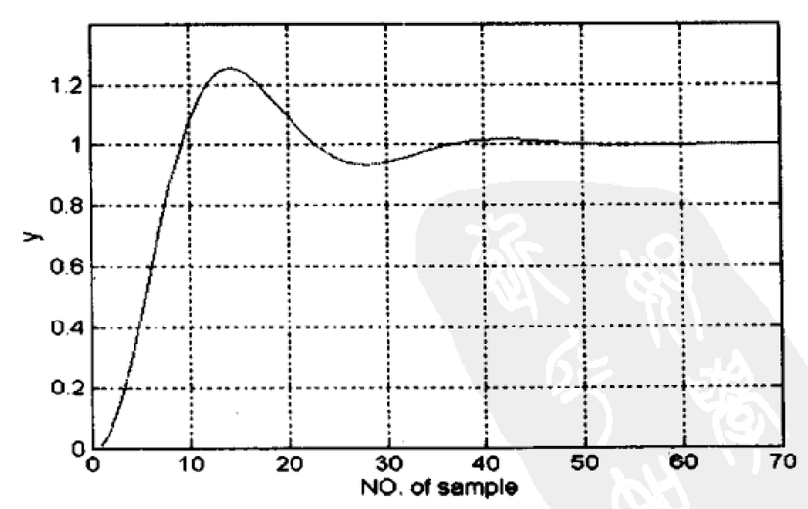


图9-16 例9-7的脉冲响应曲线

【例9-8】离散系统传递函数为

$$G(z)=\frac{0.4673z^{-1}-0.3393z^{-2}}{1-1.5327z^{-1}+0.6607z^{-2}}=\frac{0.4673z-0.3393}{z^{2}-1.5327z+0.6607}$$

输入为单位阶跃响应，$u(k)=1$。

令输入形式为u=ones(1,41)

MATLAB程序9-10是求取系统阶跃响应的程序清单：

```
% MATLAB PROGRAM 9-10
num=[0 0.4673 -0.3393];den=[1 -1.5327 0.6607];
u=ones(1,41);k=0:40;
y=filter(num,den,u);
plot(k,y,'g-'),grid
xlabel('k'),ylabel('y(k)')
end
```

其响应曲线见图9-17。

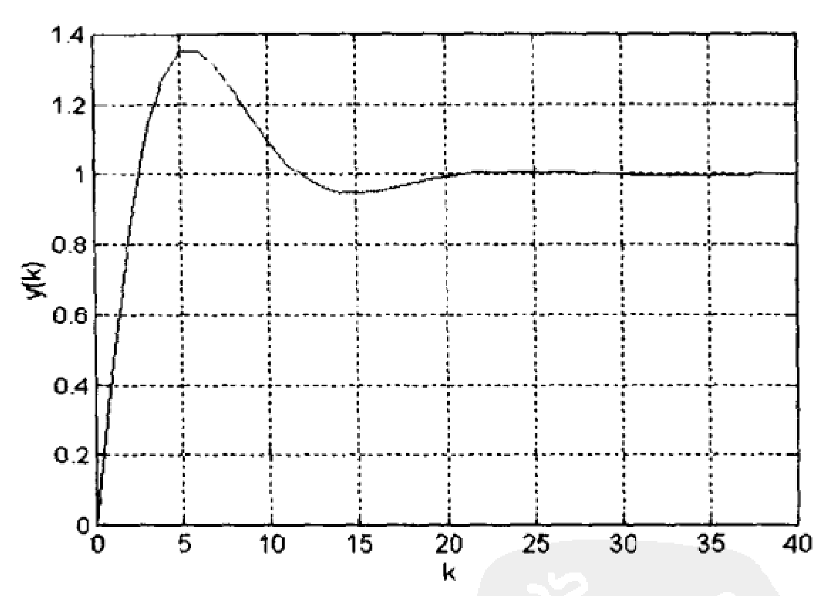


图9-17　例9-8的阶跃响应曲线

其特征方程为

$$z^{2}-1.5327z+0.6607$$

则其　den=〔1,-1.5327,0.6607〕

其特征方程的根为

r=roots(den)

```
r=
  0.7664+0.2709i
  0.7664-0.2709i
```

【例9-9】系统传递函数为

$$\frac{Y(z)}{X(z)}=\frac{0.7870z}{z^2-0.8195z+0.6065}$$

输入为斜坡输入信号，采样周期T=0.5s，绘制单位斜坡响应曲线。

令斜坡输入$u=kT$ （k=0, 1, 2, …, 20)，则在MATLAB程序中u=(k*T)=(k*0.5)，k=(0:20)

在斜坡响应中，采样周期T是很重要的。

MATLAB程序9-11是产生斜坡响应曲线的程序清单：

```
% MATLAB PROGRAM 9-11
num=[0 0.7870 0];den=[1 -0.8195 0.6065];
k=0:20;u=[0.5*k];
y=filter(num,den,u);
plot(k,y,'r-',k,0.5*k,'g-'),grid
xlabel('k'),ylabel('y(k)')
end
```

其响应曲线见图9-18。

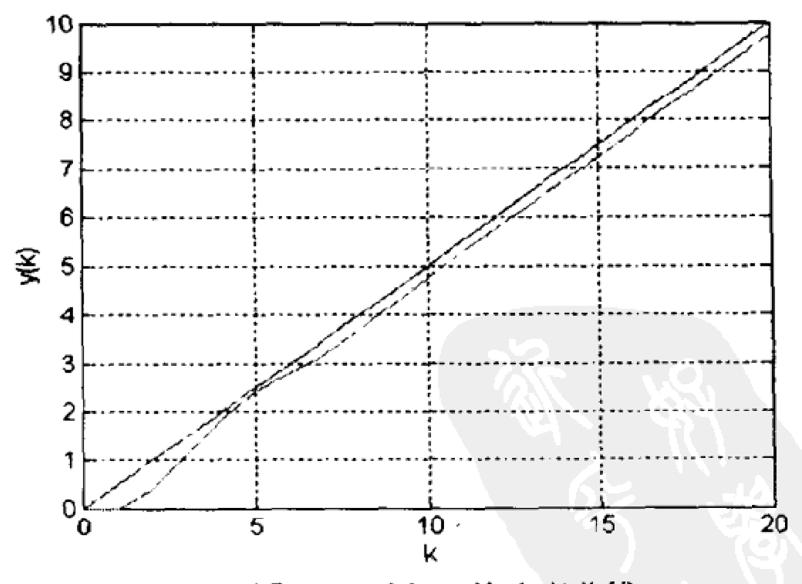


图9-18 例9-9的响应曲线

求输入斜坡响应曲线还有一些其它方法，如将传递函数乘

以 $Tz^{-1}/(1-z^{-1})^2$，然后求取脉冲响应，两种方法的结果是一致的。

9.3.3 具有离散系统状态空间方程的系统动态响应设计实例

离散系统的状态空间方程为

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)$$

$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)+\boldsymbol{D}\boldsymbol{u}(k)$$

首先，可以用下列命令将状态空间方程变换成脉冲传递函数。

(num,den)=ss2tf(G,H,C,D)

然后用如下命令求取系统的响应曲线

y=filter(num,den,u)

如果想得到状态矢量 $x_1(k)$ 的响应，用如下方程：

$$x_1(k)=\begin{bmatrix}1, & 0\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix}=\boldsymbol{F}\boldsymbol{x}(k)$$

式中，$\boldsymbol{F}$=[1, 0]，则状态空间方程转换为脉冲传递函数的命令为

[num1,den1]=ss2tf(G,H,F,D)

然后使用　x1=filter(num1,den1,u)

同样得到 $x_2(k)$ 的响应为

$$x_2(k)=\begin{bmatrix}0, & 1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix}=\boldsymbol{J}\boldsymbol{x}(k)$$

式中，$\boldsymbol{J}$=[0, 1]。然后使用

[num2,den2]=ss2tf(G,H,J,D)

x2=filter(num2,den2,u)

从而得到状态变量 $x_2(k)$ 的响应

【例9-10】 对于如下状态空间方程：

$$\begin{bmatrix}x_1(k+1)\\x_2(k+1)\end{bmatrix}=\begin{bmatrix}0 & 1\\-0.16 & -1\end{bmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix}+\begin{bmatrix}0\\1\end{bmatrix}u(k)$$

$$y(k)=\begin{pmatrix}1.16 & 1\end{pmatrix}\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix}$$

求系统的阶跃响应。

MATLAB程序9-12是求取系统阶跃响应的程序清单。

```
% MATLAB PROGRAM 9-12
```

```
G=[0 1;-0.16 -1];H=[0;1];
C=[1.16 1];D=[0];
F=[1 0];J=[0 1];
[num,den]=ss2tf(G,H,C,D);
k=0:50;u=ones(1,51);
y=filter(num,den,u);
plot(k,y,'g-'),grid
xlabel('k');ylabel('y(k)')
axis([0 30 0 1.2]);
[num1,den1]=ss2tf(G,H,F,D);
x1=filter(num1,den1,u);
figure;plot(k,x1,'g-'),grid
xlabel('k');ylabel('x1(k)')
axis([0 30 -0.3 1.3]);
[num2,den2]=ss2tf(G,H,J,D);
x2=filter(num2,den2,u);
figure;plot(k,x2,'g-'),grid
xlabel('k');ylabel('x2(k)')
axis([0 30 -0.3 1.3]);
end
```


其输出响应曲线见图9-19。

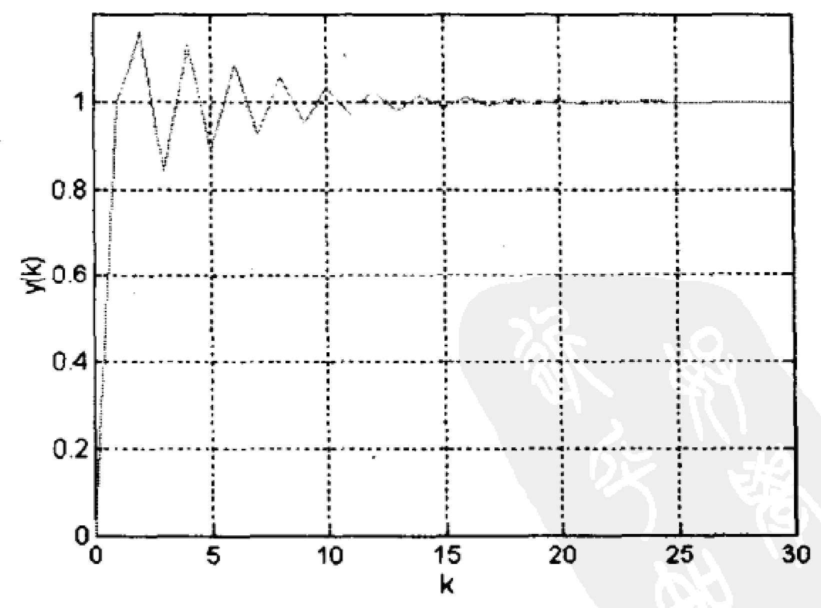


图9-19　单位阶跃响应曲线

状态变量$x_1(k)$、$x_2(k)$的响应曲线见图9-20、图9-21。

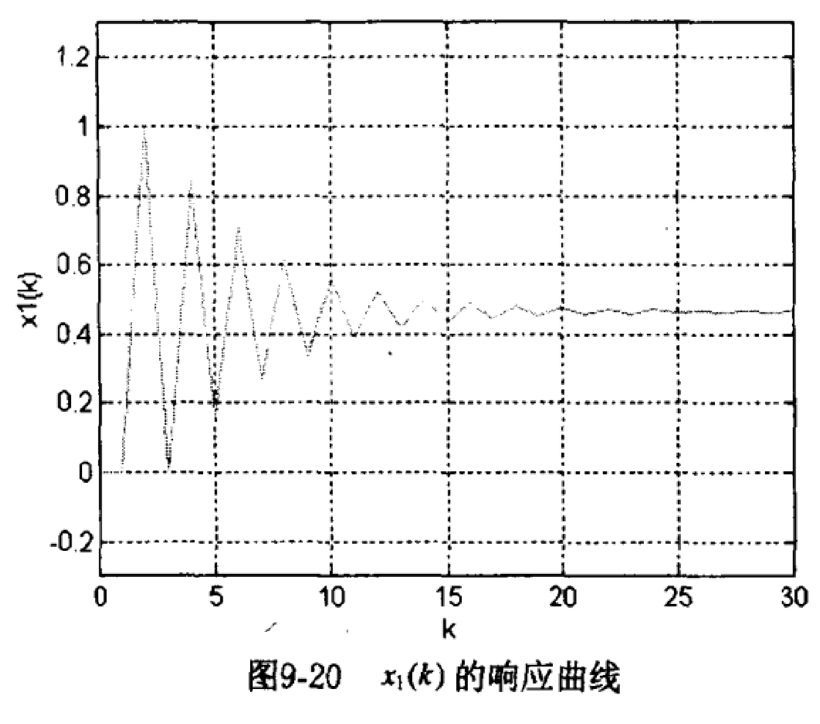


图9-20 $x_1(k)$ 的响应曲线

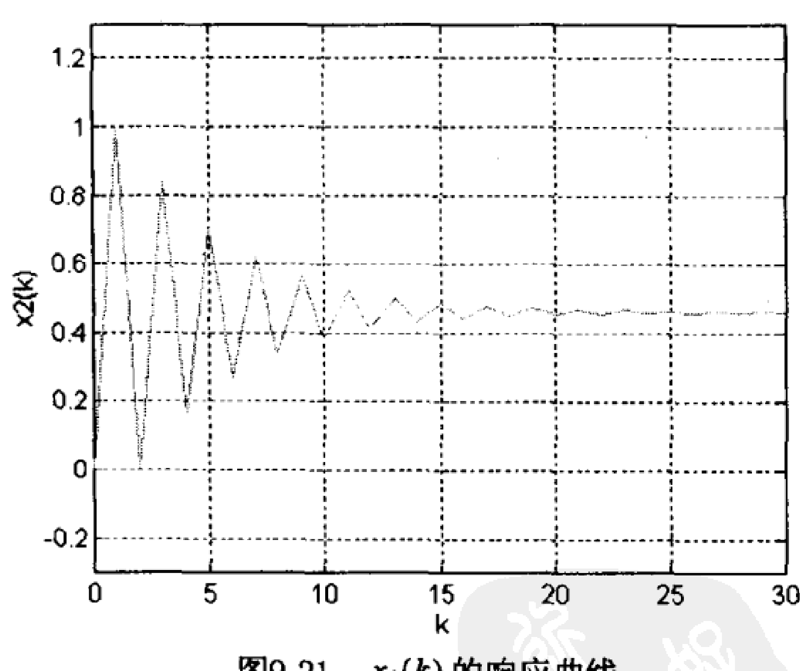


图9-21 $x_2(k)$ 的响应曲线

9.3.4 离散控制系统的频率响应

z变换将s平面左半平面变换成单位圆内，因为传统的频率响应方法适用于整个左半平面，因此不能应用到z平面上。

为了克服这种困难，可以用双线性变换将z平面的传递函数变换到ω平面，这种变换叫做ω变换。

$$z=\frac{1+(T/2)\omega}{1-(T/2)\omega}$$

式中，T是离散系统的采样周期。将z平面的传递函数变换成ω平面的函数，以前介绍的频率响应方法可以应用到离散系统中。

【例9-11】 采样周期T=0.1s，传递函数为

$$G(z)=\frac{10}{s+10}$$

将其变换到z平面。

其MATLAB程序如下：

```
num=[0 10];
den=[1 10];
[A,B,C,D]=tf2ss(num,den);
[G,H]=c2d(A,B,0.1);
[numz,denz]=ss2tf(G,H,C,D)
end
```

上述变换采用零阶保持器法，由此得

$$G(z)=\frac{0.6321}{z-0.3679}$$

用伯德图设计数字控制器包括两次不同的双线性变换。首先，将$G(z)$变换到$G(\omega)$，其变换式为

$$z=\frac{1+(T/2)\omega}{1-(T/2)\omega}$$

然后，在ω平面设计数字控制器$G_c(\omega)$。再将$G_c(\omega)$变换到z平面的$G_c(z)$，其变换式为

$$\omega=\frac{2}{T}\cdot\frac{z-1}{z+1}$$

可以用MATLAB语言中c2dm命令的双线性变换方法将$G(\omega)$变换成$G(z)$，却不能用c2dm命令将$G(z)$变换成$G(\omega)$。因此必须做某些修改，看下面的例子。

【例9-12】 将$G(z)=0.6321/(z-0.3679)$变换到ω平面上，T=0.1s。

z到ω的变换为

$$z=\frac{1+(T/2)\omega}{1-(T/2)\omega}=\frac{1+0.05\omega}{1-0.05\omega}$$

令$v=-0.05\omega$

则
$$z=\frac{1-v}{1+v}=(-1)\cdot\frac{v-1}{v+1}$$

双线性变换采用的公式为

$$z=\frac{2}{T}\cdot\frac{v-1}{v+1}=\frac{v-1}{v+1}\bigg|_{T=2}$$

比较上述两式，可以采用如下步骤将G(z)变换到G(ω)。

(1)在$G(z)$式中，用$-z$代替z，则传递函数变为

```
num=[0,0.6321]
den=[-1,-0.3679]
```

(2)用下列转换命令：

```
[numv,denv]=c2dm(num,den,ts,'t')
```

若式中ts=2，则

```
numv=[-0.4621   -0.4621]
denv=[1.0000    -0.4621]
```

即
$$G(v)=\frac{\text{numv}}{\text{denv}}=\frac{-0.4621v-0.4621}{v-0.4621}$$

(3)用$v=-0.05\omega$代入$G(v)$，从而得到$G(\omega)$

$$G(\omega)=\frac{-0.4621(-0.05\omega)-0.4621}{-0.05\omega-0.4621}$$

在MATLAB程序中，上式的计算可以用如下程序完成：

```
numw=[-0.4621   -0.4621].*[-0.05   1]
denw=[1          -0.4621].*[-0.05   1]
```

MATLAB程序9-13是完成以上各部分的程序清单。

```
% MATLAB PROGRAM 9-13
num=[0 0.6321];den=[-1 -0.3679];
ts=2;
[numv denv]=c2dm(num,den,ts,'t');
numw=[numv].*[1 -20]
denw=[denv].*[1 -20]
end
```

其结果为

```
numw =
        -0.4621      9.2419
denw =
         1.0000      9.2419
```

则$G(\omega)$为

$$G(\omega)=\frac{-0.4621\omega+9.2419}{\omega+9.2419}$$

【例9-13】 对图9-22所示的数字控制系统，在ω平面设计一数字控制器，使系统的相角裕量为50°，幅值裕量至少为10dB，稳态速度误差常数K_v=2，采样周期T=0.2s。

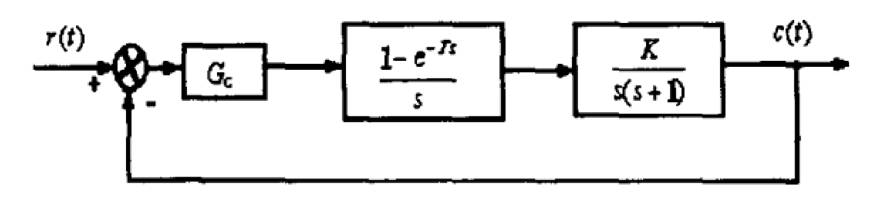


图9-22 控制系统结构框图

解 系统的脉冲传递函数为

$$G(z)=Z\left[\frac{1-e^{-Ts}}{s}\cdot\frac{K}{s(s+1)}\right]$$

$$=\frac{K(0.01873z+0.01752)}{z^2-1.8187z+0.8187}$$

为了使用伯德图设计方法，必须将$G(z)$转变为ω平面的传递函数$G(\omega)$，采用如下变换式可以得到$G(\omega)$：

$$z=\frac{1+(T/2)\omega}{1-(T/2)\omega}=\frac{1+0.1\omega}{1-0.1\omega}$$

则

$$G(\omega)=\frac{K(-0.000333\omega^2-0.09633\omega+0.9966)}{\omega^2+0.9969\omega}$$

$$\approx\frac{K(1+\omega/300)(1-\omega/10)}{\omega(\omega+1)}$$

上式给出了转折频率的位置，其中一个零点在ω的右半平面(ω=10)，因此，$G(\omega)$是非最小相位系统。

假设数字控制器$G_c(\omega)$的传递函数在低频段有单位增益，并有如下的形式：

$$G_c(\omega)=\frac{1+\tau\omega}{1+\alpha\tau\omega}$$

开环传递函数为

$$G_c(\omega)G(\omega)=\frac{1+\tau\omega}{1+\alpha\tau\omega}\cdot\frac{K(-0.000333\omega^2-0.09633\omega+0.9966)}{\omega^2+0.9969\omega}$$

稳定速度误差常数$K_v=2$，因此

$$K_v=\lim_{\omega\to 0}\omega G_c(\omega)G(\omega)\approx K=2$$

则

$$G(\omega)=\frac{-0.000666\omega^2-0.19266\omega+1.9932}{\omega^2+0.9969\omega}$$

MATLAB程序9-13是绘制$G(\omega)$的伯德图的程序清单。

```
% MATLAB PROGRAM 9-13
num=[-0.000666 -0.19266 1.9932];den=[1 0.9969 0];
w=logspace(-1,3,100);
bode(num,den,w);
end
```

$G(\omega)$的伯德图见图9-23。

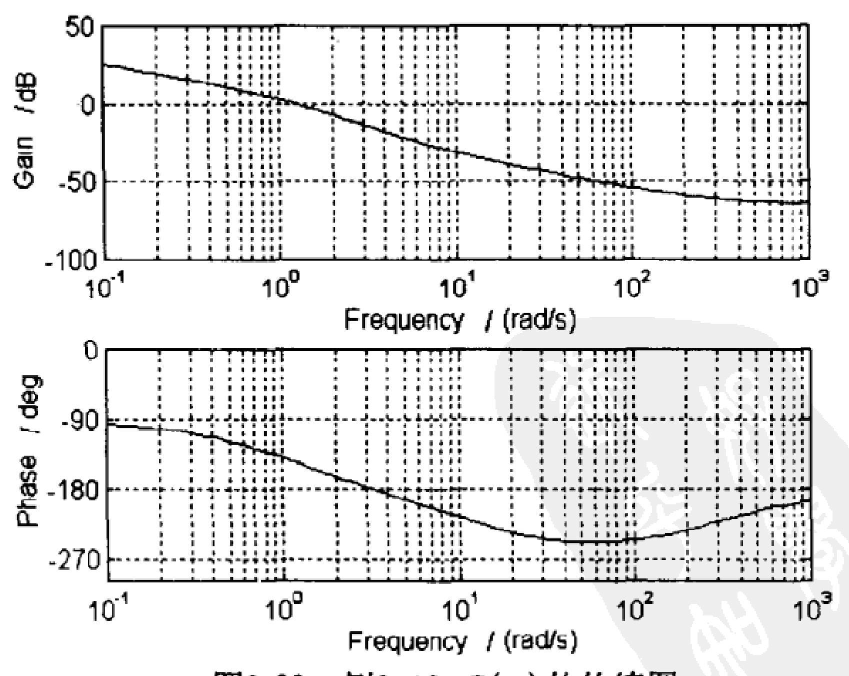


图9-23 例9-13 $G(\omega)$的伯德图

从其伯德图上可见，相角裕量为30°，幅值裕量为14.5dB，而要求的相角裕量为50°，幅值裕量为10dB。我们将采用传统的设计方法设计数字控制器，以满足上述要求。

对超前补偿设计，因为要求相角裕量为50°，满足这些要求的附加超前相位角为20°，为达到相角裕量为50°，同时又不减小K值的目的，超前补偿器必须提供所要求的超前角。

加入超前补偿器，不改变幅频曲线的形状，只是增益穿越频率向右移动了，考虑到幅频穿越频率的移动，假设要求的最大超前角将近为28°（也就是说，由于穿越频率的右移必须在补偿器上多增加8°的相角裕量）。由于

$$\sin\varphi_{m}=\frac{1-\alpha}{1+\alpha}$$

由 φ_m=28°，可以推出α=0.361。一旦α被决定后，下一步就是求取超前补偿器的转折频率$\omega=1/T$和$\omega=1/(\alpha T)$。首先，要注意，其最大超前角是在两个转折频率的几何平均值上，$\omega=1/(\sqrt{2}T)$。在幅频曲线上的$\omega=1/(\sqrt{2}T)$点，由于加入$(1+T\omega)/[1+\alpha(\omega)]$，其修正的量是

$$\left|\frac{1+\tau\omega}{1+\alpha\tau\omega}\right|_{\omega=1/(\sqrt{\alpha}\tau)}=\frac{1}{\sqrt{\alpha}}$$

然后，求取补偿系统的幅值等于$-20\log(1/\sqrt{2})$的频率点，由于

$$-20\log(1/\sqrt{0.361})=-20\log1.6643=-4.425\,\text{dB}$$

为了求取幅值等于－4.425dB的频率点，绘制在$1\leqslant\omega\leqslant10$rad/s频率范围的伯德图。MATLAB程序9-14是求取系统伯德图的程序清单。

```
% MATLAB PROGRAM 9-14
num=[-0.000666 -0.19266 1.9932];
den=[1 0.9969 0];
w=logspace(0,1,100);
bode(num,den,w);
end
```

其伯德图见图9-24。从该图上可以求取幅值为-4.4dB的频

率点为ω=1.7rad/s。

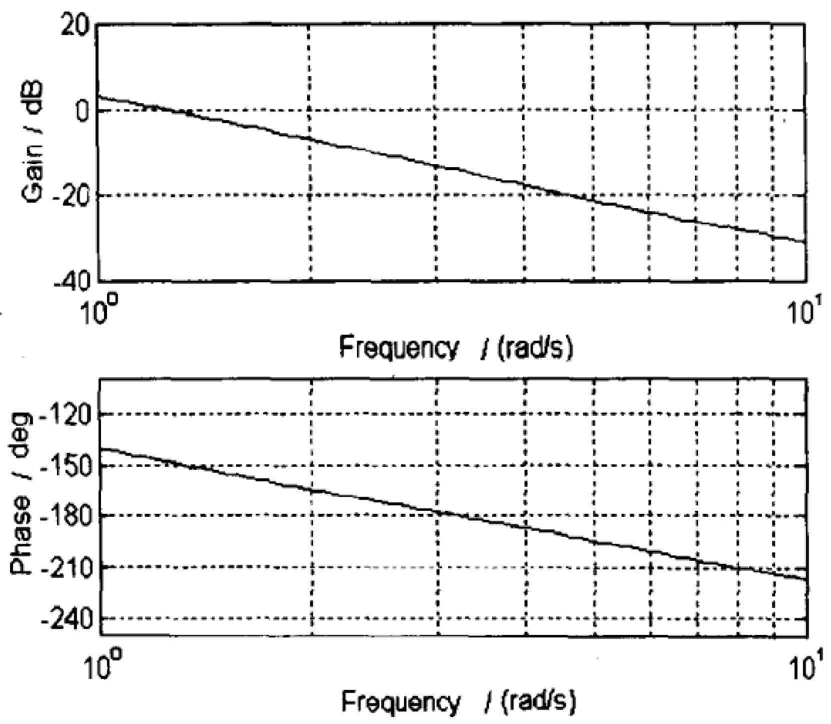


图9-24　系统的伯德图

选取该频率点为新的幅值穿越频率ω_c，则该点频率对应于$1/(\sqrt{\alpha}\,\tau)$，即

$$\omega_c = 1/(\sqrt{\alpha}\,\tau) = 1.7$$

则　$\tau = 1/(1.7\sqrt{\alpha}) = 0.9790 \quad \alpha\tau = 0.3534$

因此超前补偿器为

$$G_c(\omega) = \frac{1+\tau\omega}{1+\alpha\tau\omega} = \frac{1+0.9790\omega}{1+0.3534\omega}$$

下面绘制超前补偿器的伯德图，MATLAB程序9-15是产生补偿器的伯德图的程序清单。

```
% MATLAB PROGRAM 9-15
numc=[0.9790 1];
denc=[0.3534 1];
w=logspace(-1,3,100);
bode(numc,denc,w);
end
```

其伯德图见图9-25。

MATLAB程序9-16是绘制$G(\omega)$、$G_c(\omega)$和$G_c(\omega)G(\omega)$的伯

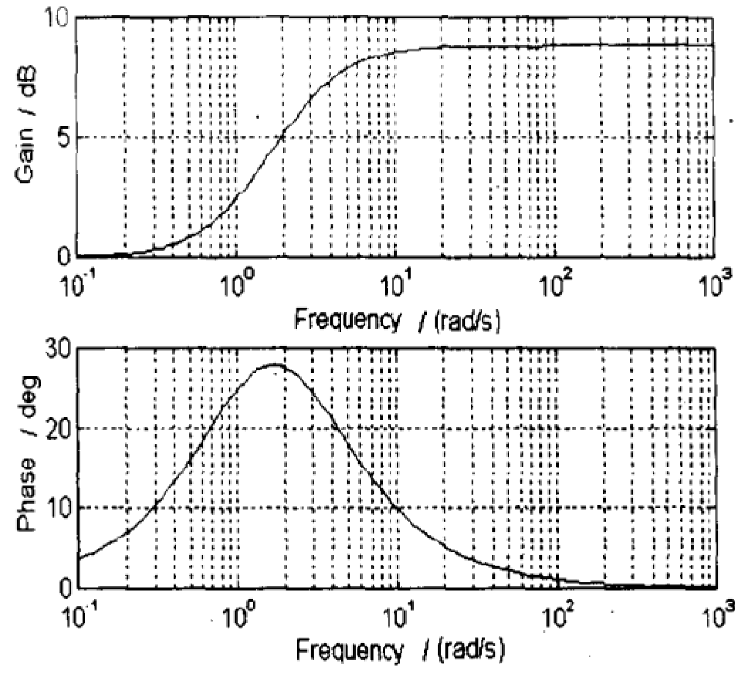


图9-25　超前补偿器的伯德图

德图的程序清单。

```
% MATLAB PROGRAM 9-16
% Bode  Diagrams
num=[-0.000666 -0.19266 1.9932];
den=[1 0.9969 0];
numc=[0.9790 1];
denc=[0.3534 1];
w=logspace(-1,3,100);
[mag,phase,w]=bode(num,den,w);
magdb=20*log10(mag);
[magc,phasec,w]=bode(numc,denc,w);
magcdb=20*log10(magc);
% magnitude Diagrams of Gc(w)G(w)
mgdb=magdb+magcdb;
semilogx(w,magdb,'ro',w,magdb,'r-',w,magcdb,'r+',...
w,magcdb,'r-',w,mgdb,'g-')
xlabel('Frequency (rad/s)');
ylabel('gain (db)');
text(10,-42,'G(w)')
```

```
text(10,14,'Gc(w)')
text(10,-20,'Gc(w)G(w)')
% phase-angle curves of Gc(w)G(w)
figure
pha=phase+phasec;
ph180=-180*ones(1,100);
semilogx(w,phase,'ro',w,phase,'r-',w,phasec,'r+',...
w,phasec,'r-',w,pha,'g-',w,ph180,'--')
xlabel('Frequency (rad/s)')
ylabel('Phase (deg)')
text(0.7,-160,'G(w)')
text(0.7,10,'Gc(w)')
text(0.7,-100,'Gc(w)G(w)')
end
```

图9-26给出了三种传递函数的幅频特性曲线，图9-27给出了三种传递函数的相频特性曲线，三种曲线被画在同一个特性图上。

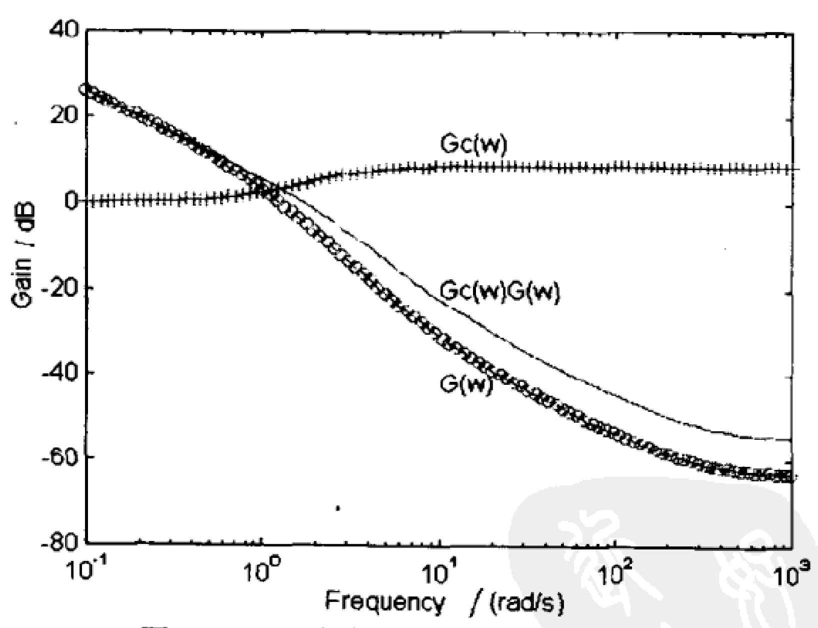


图9-26　三种传递函数的幅频特性曲线

从$G_c(\omega)G(\omega)$的幅频特性和相频特性可以看出，幅值裕量近似为14dB，相角裕量近似为50°。

因为K满足速度误差系数K_v的要求，加入补偿后的系统性能指标满足要求，因此所设计的数字补偿器是符合要求的。

对于同一个设计系统，其结果不是唯一的，许多其它的数字补偿器$G_c(\omega)$也可能满足系统所要求的性能指标。

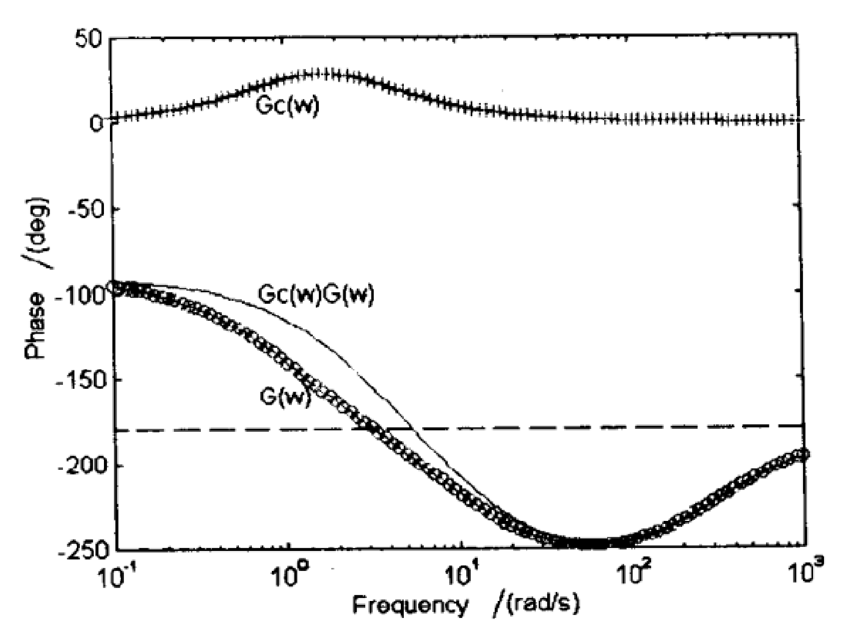


图9-27 三种传递函数的相频特性曲线

上述在ω平面设计的数字控制器，应该用c2dm变换方法转换到z平面上。

$$\omega=\frac{2}{T}\cdot\frac{z-1}{z+1}$$

$$G_c(z)=G_c(\omega)\bigg|_{\omega=\frac{2}{0.2}\frac{z-1}{z+1}}=\frac{1+0.9790\omega}{1+0.3534\omega}\bigg|_{\omega=10\frac{z-1}{z+1}}$$

上述变换可以用如下命令来完成：

```
numw=[1 0.9790];denw=[1 0.3534];
[numz,denz]=c2dm(numw, denw, 0.2,'t')
```

从而得出

```
numz=[2.3798   -1.9387]
denz=[1.0000   -0.5589]
```

则在z平面的传递函数$G_c(z)$为

$$G_c(z)=\frac{2.3798z-1.9387}{z-0.5589}$$

加入补偿器后的系统开环传递函数为

$$G_c(z)G(z)=\frac{2.3798z-1.9387}{z-0.5589}\cdot\frac{0.03746z+0.03504}{z^2-1.8187z+0.8187}$$

两个多项式相乘可以用conv命令求取结果如下：

```
a=[2.3798 -1.9387];
b=[0.03746 0.03504];
p=conv(a, b)
c=[1 -0.5589];
d=[1 -1.8187 0.8187];
q=conv(c, d)
```

由此得

$$G_c(z)G(z)=\frac{0.0891z^2+0.0108z-0.0679}{z^3-2.3776z^2+1.8352z-0.4576}$$

系统的闭环传递函数为

$$\frac{C(z)}{R(z)}=\frac{0.0891z^2+0.0108z-0.0679}{z^3-2.2885z^2+1.8460z-0.5255}$$

闭环系统的零极点可以用如下程序得到：

```
num=[0 0.0891 0.0108 -0.0679];
den=[1 -2.2885 1.8460 -0.5255];
roots(num)
roots(den)
```

其零点为

```
ans =
   -0.9357
    0.8145
```

其极点为

```
ans =
     0.8126
     0.7379 + 0.3196i
     0.7379 - 0.3196i
```

闭环传递函数有两个零点，分别为z=-0.9357、z=0.8145，而z=0.8145的零点几乎与闭环系统的极点p=0.8126相抵消，另一个零点z=-0.9357对系统的动态性能及频率响应影响很小，因为它是在z平面的负实轴上，在0～-1之间靠近-1点的位置。

MATLAB程序9-17是求取系统动态响应曲线的程序清单。

```
% MATLAB PROGRAM 9-17
```

```
num=[0 0.0891 0.0108 -0.0679];
den=[1 -2.2885 1.8460 -0.5255];
x=ones(1,41);
k=0:40;
y=filter(num,den,x);
plot(k,y,'-')
grid
xlabel('k')
ylabel('Output y(k)')
v=[0 40 0 1.4];
axis(v)
end
```

系统的单位阶跃响应曲线见图9-28。

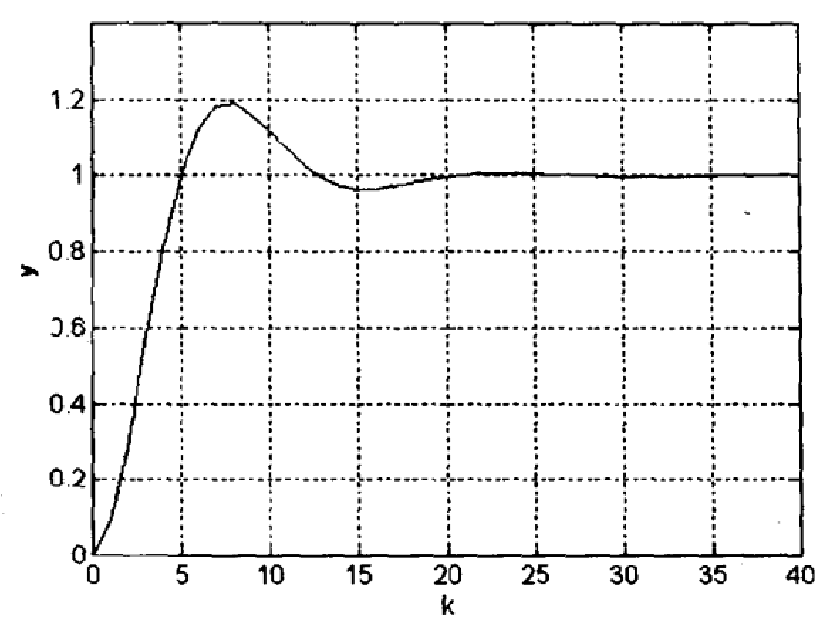


图9-28 系统的单位阶跃响应曲线

从图9-28可见，系统的最大超调量为20%，过渡过程时间为4s。

假如没有数字补偿器，也不加零阶保持器，只是增益被调整至2，系统的闭环传递函数为

$$\frac{C(s)}{R(s)}=\frac{2}{s^2+s+2}$$

系统的最大超调量为30%，过渡过程时间为8s。

由上面可以看出，加入超前补偿器大大地改善了系统的动态性能指标。

ω变换的优点是使传统的伯德图设计系统频率响应的方法，可以用于离散系统的设计。

ω变换可能会使$G(\omega)$在ω右半平面产生零点，假如在ω右半平面存在零点，则$G(\omega)$就是非最小相位系统。由于右半平面的零点是由采样－保持过程所产生的，所以零点的位置取决于采样周期T，当采样周期变得很小时，在右半平面的零点对系统性能的影响会变得很小。

9.3.5 离散系统观测器设计实例

【例9-14】状态观测器的设计。

考虑如下系统：

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k)$$
$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k)$$

其中

$$\boldsymbol{G}=\begin{bmatrix}0 & 0 & 1\\ 1 & 0 & 0\\ -0.2 & -0.5 & 1.1\end{bmatrix}\quad \boldsymbol{H}=\begin{bmatrix}0\\0\\1\end{bmatrix}\quad \boldsymbol{C}=\begin{bmatrix}1 & 0 & 0\end{bmatrix}$$

系统框图见图9-29，即系统采用状态反馈和最小阶观测器，因此控制信号是

$$\boldsymbol{u}(k)=-\boldsymbol{K}\tilde{\boldsymbol{x}}(k)$$

决定状态反馈增益矩阵$\boldsymbol{K}$，要求系统对任何初始值均为无差拍响应。假设输出$\boldsymbol{y}(k)$是可测量的，设计最小阶观测器，使观测器误差响应是无差拍的。

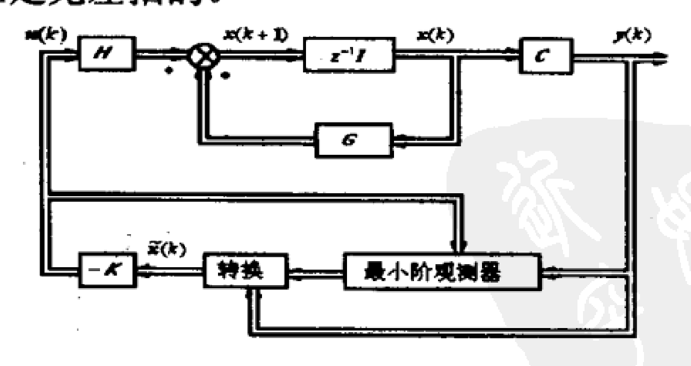


图9-29　例9-14系统框图

MATLAB程序9-18是求取状态反馈增益矩阵$\boldsymbol{K}$和最小阶观测器增益矩阵$\boldsymbol{K}_e$的程序清单。

```
% MATLAB PROGRAM  9-18
% Design of a control system
%**In this program determine the state feedback
gain
% matrix K and min-order observer gain matrix Ke
%**Enter matrices G,H,C**
G=[0 0 1;1 0 0;-0.2 -0.5 1.1];
H=[0;0;1];C=[1 0 0];
M=[H G*H G^2*H];
rank(M)
% Since rank of M is 3, Hence arbitrary pole
% placement are possible
Phi=G^3;
K=[0 0 1]*inv(M)*Phi
% Next design a min-order observer
Gaa=[G(1,1)];
Gab=[G(1,2) G(1,3)];
Gba=[G(2,1);G(3,1)];
Gbb=[G(2,2) G(2,3);G(3,2) G(3,3)];
Ha=[H(1,1)];
Hb=[H(2,1);H(3,1)];
NN=[Gab' Gbb'*Gab'];
rank(NN)
% Since rank of N is 3, Hence design of
% min-order observer is possible
% placement are possible
Ph=Gbb^2;
Ke=Ph*inv(NN')*[0;1]
end
```

观测器增益矩阵$\boldsymbol{K}_e$为

$$\boldsymbol{K}_e=\begin{bmatrix}0\\1.1\end{bmatrix}$$

下面验证一下所设计系统的响应对于所有的初始条件是无差拍的。对最小阶观测器系统，状态反馈控制系统的方程为

$$\boldsymbol{x}(k+1)=\boldsymbol{G}\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{u}(k) \tag{9-7}$$

$$\boldsymbol{y}(k)=\boldsymbol{C}\boldsymbol{x}(k) \tag{9-8}$$

$$\boldsymbol{u}(k)=-\boldsymbol{K}\widetilde{\boldsymbol{x}}(k) \tag{9-9}$$

状态矢量 $\boldsymbol{x}(k)$ 和估测的状态矢量 $\widetilde{\boldsymbol{x}}(k)$ 可被写为

$$\boldsymbol{x}(k)=\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}=\begin{bmatrix} x_1(k) \\ \cdots\cdots \\ \boldsymbol{x}_{\mathrm{b}}(k) \end{bmatrix}$$

$$\widetilde{\boldsymbol{x}}(k)=\begin{bmatrix} x_1(k) \\ \widetilde{x}_2(k) \\ \widetilde{x}_3(k) \end{bmatrix}=\begin{bmatrix} x_1(k) \\ \cdots\cdots \\ \widetilde{\boldsymbol{x}}_{\mathrm{b}}(k) \end{bmatrix}$$

由以上方程可得

$$\begin{aligned} \boldsymbol{x}(k+1)&=\boldsymbol{G}\boldsymbol{x}(k)-\boldsymbol{H}\boldsymbol{K}\widetilde{\boldsymbol{x}}(k) \\ &=(\boldsymbol{G}-\boldsymbol{H}\boldsymbol{K})\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{K}(\boldsymbol{x}(k)-\widetilde{\boldsymbol{x}}(k)) \end{aligned} \tag{9-10}$$

式中

$$\begin{aligned} \boldsymbol{x}(k)-\widetilde{\boldsymbol{x}}(k)&=\begin{bmatrix} x_1(k) \\ \cdots\cdots \\ \boldsymbol{x}_{\mathrm{b}}(k) \end{bmatrix}-\begin{bmatrix} x_1(k) \\ \cdots\cdots \\ \widetilde{\boldsymbol{x}}_{\mathrm{b}}(k) \end{bmatrix} \\ &=\begin{bmatrix} 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \boldsymbol{x}_{\mathrm{b}}(k)-\widetilde{\boldsymbol{x}}_{\mathrm{b}}(k) \end{bmatrix}=\begin{bmatrix} 0 \\ \cdots \\ \boldsymbol{e}(k) \end{bmatrix} \end{aligned}$$

其中

$$\boldsymbol{e}(k)=\boldsymbol{x}_{\mathrm{b}}(k)-\widetilde{\boldsymbol{x}}_{\mathrm{b}}(k)=\begin{bmatrix} e_1(k) \\ e_2(k) \end{bmatrix}$$

定义

$$\boldsymbol{F}=\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

则
$$\boldsymbol{x}(k)-\widetilde{\boldsymbol{x}}(k)=\boldsymbol{F}\boldsymbol{e}(k)$$

于是上述方程可写为

$$\boldsymbol{x}(k+1)=(\boldsymbol{G}-\boldsymbol{H}\boldsymbol{K})\boldsymbol{x}(k)+\boldsymbol{H}\boldsymbol{K}\boldsymbol{F}\boldsymbol{e}(k) \tag{9-11}$$

前面所介绍的观测器误差方程是

$$\boldsymbol{e}(k+1)=(\boldsymbol{G}_{\mathrm{bb}}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}})\boldsymbol{e}(k) \tag{9-12}$$

将方程式(9-11)与式(9-12)结合，则最小阶状态观测器的系统控制程是

$$\begin{bmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{e}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}-\boldsymbol{HK} & \boldsymbol{HKF} \\ \boldsymbol{0} & \boldsymbol{G}_{\mathrm{bb}}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{e}(k) \end{bmatrix} = \boldsymbol{G}_1 \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{e}(k) \end{bmatrix} \quad (9\text{-}13)$$

式中

$$\boldsymbol{G}_1 = \begin{bmatrix} \boldsymbol{G}-\boldsymbol{HK} & \boldsymbol{HKF} \\ \boldsymbol{0} & \boldsymbol{G}_{\mathrm{bb}}-\boldsymbol{K}_{\mathrm{e}}\boldsymbol{G}_{\mathrm{ab}} \end{bmatrix}$$

由上述方程式(9-13)可得

$$\begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{e}(k) \end{bmatrix} = \boldsymbol{G}_1^{\,k} \begin{bmatrix} \boldsymbol{x}(0) \\ \boldsymbol{e}(0) \end{bmatrix} \qquad (k=1,\ 2,\ 3,\ \cdots)$$

如果设计系统确实是无差拍系统，则对任何初始条件，过渡过程时间至多为5个采样周期。

MATLAB程序9-19给出了$\boldsymbol{G}_1$、$\boldsymbol{G}_1^{2}$、…、$\boldsymbol{G}_1^{5}$的值。

```
% MATLAB PROGRAM 9-19
% computation of G1^k, where k=1,2,3,4,5
F=[0 0;1 0;0 1];
G1=[G-H*K H*K*F;zeros(2,3) Gbb-Ke*Gab]
G1^2,G1^3,G1^4,G1^5
end
```

$\boldsymbol{G}_1^{5}$是5×5的零矩阵。

对于任意的初始条件$\begin{bmatrix} \boldsymbol{x}(0) \\ \boldsymbol{e}(0) \end{bmatrix}$，其响应在5个周期内变为0。其响应是无差拍响应，即所设计的系统满足条件要求。

第 10 章　常用 MATLAB 工具箱简介

由于 MATLAB 的使用非常方便，且提供了数值计算、图形绘制、数据处理、图像处理等方面的丰富可靠的功能，因此许多控制领域的专家在 MATLAB 的基础上将他们擅长的领域编写了一些专用的应用程序集，称之为工具箱，如控制系统工具箱（Control System Toolbox）、信号处理工具箱（Signal Processing Toolbox）、系统辨识工具箱（System Identification Toolbox）、鲁棒控制工具箱（Robust Control Toolbox）、多变量频域设计工具箱（Multivariable Frequency Design Toolbox）、多变量系统分析与综合工具箱（Mu-analysis and Synthesis Toolbox）、非线性控制工具箱（Nonlinear Control Toolbox）、神经网络工具箱（Neural Network Toolbox）、模糊控制工具箱（Fuzzy Control Toolbox）等。由于参与编写这些工具箱的设计者包括许多国际控制领域的著名学者，因此提高了 MATLAB 的声誉和可信度，为 MATLAB 的应用打开了崭新的局面。

本章将简要地介绍几种常用的 MATLAB 工具箱。

10.1　控制系统工具箱

控制系统工具箱用于实现一般的控制系统设计、分析与建模。它用传递函数和状态方程等形式进行系统建模，包括连续时间系统和离散时间系统。它还提供了各个模型之间的转换、时域和频域的控制系统分析与设计方法。

控制系统工具箱中的许多命令在本书的前几章都有介绍。下面就按其功能分类进行一些简要介绍。

控制系统工具箱就其功能来说，可以分成以下几个部分：

(1) 模型建立；

(2) 模型转换；

(3) 模型降阶；

(4) 模型实现；

(5) 模型性质；

(6) 时域响应；

(7) 频域响应；

(8) 根轨迹；

(9) 增益选择；

(10) 方程求解；

(11) 演示程序；

(12) 一些实用程序。

下面分别简要地介绍各部分主要命令的功能。

10.1.1 模型建立

建立模型时可采用 append、augstate、blkbuild、cloop、connect、conv、destim、dreg、drmodel、estim、feedback、ord2、pade、parallel、reg、rmodel、series、ssdelete、ssselect 等命令。

控制系统工具箱主要处理的是线性时不变系统（Linear Time Invariant —— LIT）。

主要应用的模型分为连续和离散两大类。而在各大类中又使用状态空间模型、传递函数模型和零极点增益模型。连续系统状态方程模型为

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t) \tag{10-1}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)$$

MATLAB 用 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$ 四个矩阵来描述此模型。

连续系统的传递函数模型为

$$G(s)=\frac{Y(s)}{U(s)}=\frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \cdots + a_{n+1}} \tag{10-2}$$

MATLAB 用 num、den 两个系统矢量分别描述 $G(s)$的分子系数

和分母系数，即

num =[b1 b2 ⋯ bm+1]

den =[a1 a2 ⋯ an+1]

连续系统的零极点增益模型为

$$G(s)=K\frac{(s+z_1)(s+z_2)\cdots(s+z_m)}{(s+p_1)(s+p_2)\cdots(s+p_n)} \quad (10\text{-}3)$$

MATLAB 用 ***Z***、***P***、*K* 三个量来描述，即

$\boldsymbol{P}=[p_1 \quad p_2 \quad \cdots \quad p_n]$

$\boldsymbol{Z}=[z_1 \quad z_2 \quad \cdots \quad z_m]$

离散系统状态方程模型为

$$\begin{aligned} \boldsymbol{x}(k+1) &= \boldsymbol{Fx}(k)+\boldsymbol{Gu}(k) \\ \boldsymbol{y}(k+1) &= \boldsymbol{Cx}(k+1)+\boldsymbol{Du}(k+1) \end{aligned} \quad (10\text{-}4)$$

MATLAB 用 ***F***、***G***、***C***、***D*** 来描述此模型。

离散系统 z 传递函数模型为

$$G(z)=\frac{f_1 z^m+f_2 z^{m-1}+\cdots+f_{m+1}}{g_1 z^n+g_2 z^{n-1}+\cdots+g_{n+1}} \quad (10\text{-}5)$$

MATLAB 也是用 num、 den 两个系数矢量分别描述 $G(s)$的分子系数和分母系数，即

num =[f1 f2 ⋯ fm+1]

den =[g1 g2 ⋯ gn+1]

离散系统的零极点增益模型的描述与连续系统时的情况类似，也是用 ***Z***、***P***、*K* 三个量来描述。

下面介绍有关模型建立的命令：

1. append 命令 将两个用状态方程表示的系统连接起来。

格式：[A,B,C,D]= append(A1,B1,C1,D1,A2,B2,C2,D2)

结果：$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{A}_1 & 0 \\ 0 & \boldsymbol{A}_2 \end{bmatrix}$ $\boldsymbol{B}=\begin{bmatrix} \boldsymbol{B}_1 & 0 \\ 0 & \boldsymbol{B}_2 \end{bmatrix}$

$$C=\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \quad D=\begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix}$$

2. augstate 命令　将系统状态增扩到输出方程中。

格式：[Ab, Bb, Cb, Db]＝augstate(A, B, C, D)

结果：$A_b=A$、$B_b=B$、$C_b=\begin{bmatrix} C \\ I \end{bmatrix}$、$D_b=\begin{bmatrix} D \\ 0 \end{bmatrix}$

3. connect 命令　将系统框图转换成状态方程模型。

格式：[aa, bb, cc, dd]＝connect(a, b, c, d, q, iu, iy)

式中，a、b、c、d 为由 append 命令建立起来的以对角方块矩阵形式给出的未连接的（A, B, C, D)系统矩阵；矩阵 q 给出了这些矩阵块的内部连接关系；矢量 iu 和 iy 给出(aa，bb，cc，dd)系统的输入和输出。

用 connect 命令进行系统建模的步骤如下：

(1) 将系统定义为若干块，块号由 1 到块数，且每块都是 SISO 的，并送入每块的分子和分母数，变量名分别为 n1、d1、n2、d2、…。

(2) 建立未连接的对角方块矩阵，实现方法如下：

1) 反复使用 tf2ss 和 append 命令，例如：

```
[a, b, c, d]= tf2ss(n1, d1);
[at, bt, ct, dt]= tf2ss(n2, d2);
[a, b, c, d]= append(a, b, c, d, at, bt, ct, dt);
[at, bt, ct, dt]= tf2ss(n3, d3);
[a, b, c, d]= append(a, b, c, d, at, bt, ct, dt);
```

2) 用 blkbuild 命令，若有 5 个块则用

```
nblocks = 5； blkbuild；
```

3) 给出内连矩阵 $\boldsymbol{q}$，内连矩阵 $\boldsymbol{q}$ 说明矩阵块之间的连接关系。每一个块占一行，每行的第一个元素说明了块号，其后的元素给出到这个块的输入情况。$\boldsymbol{q}(i, j+1)$说明了第 j 块输出到第 i 块输入之间的连接关系。

4) 给出输入和输出信息，iu 和 iy 说明了内部各块与外部输入和输出之间的关系。

5) 用 connect 命令连接成最终系统。

4. blkbuild 命令　由传递函数框图构成对角线状态方程结构。

格式：blkbuild

需用到的参数为 nblocks，n1，d1，n2，d2，…，使用方法参见 connect 命令。

5. parallel 命令　将两个系统并联连接，$u_1 = u_2$，$y = y_1 + y_2$。

格式：[A,B,C,D]= parallel(A1,B1,C1,D1,A2,B2,C2,D2)

[num, den]= parallel (num1, den1, num2, den2)

结果：$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & 0 \\ 0 & \boldsymbol{A}_2 \end{bmatrix}$　　$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 & 0 \\ 0 & \boldsymbol{B}_2 \end{bmatrix}$

$\boldsymbol{C} = [\boldsymbol{C}_1 \quad \boldsymbol{C}_2]$　　$\boldsymbol{D} = [\boldsymbol{D}_1 \quad \boldsymbol{D}_2]$

6. series 命令　将两个系统串联连接，$u_2 = y_1$，系统输入、输出分别为 u_1、y_2。

格式：[A, B, C, D]= series(A1, B1, C1, D1, A2, B2, C2, D2]

[num,den]= series(num1,den1,num2,den2)

结果：$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & 0 \\ \boldsymbol{B}_2\boldsymbol{C}_1 & \boldsymbol{A}_2 \end{bmatrix}$　　$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_1 \\ \boldsymbol{B}_2\boldsymbol{D}_1 \end{bmatrix}$

$\boldsymbol{C} = [\boldsymbol{D}_2\boldsymbol{C}_1 \quad \boldsymbol{C}_2]$　　$\boldsymbol{D} = [\boldsymbol{D}_2 \quad \boldsymbol{D}_1]$

7. feedback 命令　将两个系统反馈连接，在负反馈的情况下，$u_1 = r - y_2$，$u_2 = y_1$。

格式：[A,B,C,D]= feedback(A1,B1,C1,D1,A2,B2,C2,D2,sign)

[num, den]= feedback(num1, den1, num2, den2, sign)

式中，sign 为反馈极性，若为正反馈，则 sign 为 1；若为负反馈，则 sign 为−1。

8. cloop 命令　求单位反馈系统的闭环传递函数或状态方

程。

格式：[A, B, C, D]= cloop(A1, B1, C1, D1, sign)

[num, den]= cloop(num1,den1, sign)

式中， sign 为反馈极性，若为正反馈，则 sign 为 1 ；若为负反馈，则 sign 为-1 。

9. ord2 命令 求二阶系统的状态方程。

格式：[A, B, C, D]= ord2(wn, Z)

式中， wn 为二阶系统的自然振荡频率； Z 为阻尼系数。二阶系统的传递函数为

$$G(s)=\frac{{\omega_n}^2}{s^2+2Z\omega_n+{\omega_n}^2}$$

10. ssdelete 命令 从给定状态方程中去掉指定的输入序列和输出序列。

格式：[Ar, Br, Cr, Dr]= ssdelete(A, B, C, D, inputs, outputs)

式中，矢量 inputs 和 outputs 分别包含要去掉的输入和输出的序号。

11. ssselect 命令 从给定状态方程中选择输入序列和输出序列。

格式：[Ae, Be, Ce, De]= ssselect(A, B, C, D, inputs, outputs)

式中，矢量 inputs 和 outputs 分别包含选择的输入和输出序列。

12. pade 命令 求时滞 T 的 n 阶 pade 逼近式。

格式：[A, B, C, D]= pade(T, n)

[num, den]= pade(T, n)

对应关系式为

$$e^{-T}\approx\frac{2-T+\dfrac{T^2}{2!}s^2-\dfrac{T^3}{3!}s^3+\cdots}{2+T+\dfrac{T^2}{2!}s^2+\dfrac{T^3}{3!}s^3+\cdots}$$

num 、 den 分别为上式中分子和分母的 s 降幂形式排列的系数矢

量。

13. rmodel 命令　产生稳定的 *n* 阶随机连续时间测试模型。

格式：[num, den]= rmodel (n)

[num, den]= rmodel (n, p)

[A, B, C, D]= rmodel (n)

[A, B, C, D]= rmodel (n, p, m)

式中，当只有参数 n 时，为产生 *n* 阶单输入、单输出模型；当参数为(n, p)时，则产生单输入、*p* 个输出的模型；当参数为(n, p, m)时，则产生 *m* 个输入、*p* 个输出的模型。

14. drmodel 命令　产生稳定的 *n* 阶随机离散时间测试模型，是 rmodel 的离散版本。其格式与 rmodel 完全相同。

格式：[num, den]= drmodel (n)

[num, den]= drmodel (n, p)

[A, B, C, D]= drmoedl (n)

[A, B, C, D]= drmodel (n, p, m)

15. estim 命令　构成连续 Kalman 滤波器。

格式：[Ae, Be, Ce, De]= estim(A, B, C, D, L)

式中，L 为 Kalman 滤波器增益矩阵。

结果：$\dot{\hat{x}}=(A-LC)\hat{x}+Ly$

$$\begin{bmatrix}\hat{y}\\ \hat{x}\end{bmatrix}=\begin{bmatrix}C\\ I\end{bmatrix}\hat{x}$$

16. destim 命令　构成离散 Kalman 滤波器。

格式：[Ae, Be, Ce, De]= destim(A, B, C, D, L)

式中，L 为 Kalman 滤波器增益矩阵。

结果：$\overline{x}(k+1)=(A-ALC)\overline{x}(k)+ALy(k)$

$$\begin{bmatrix}\hat{y}(k)\\ \hat{x}(k)\end{bmatrix}=\begin{bmatrix}C-CLC\\ I-LC\end{bmatrix}\overline{x}(k)+\begin{bmatrix}CL\\ L\end{bmatrix}y(k)$$

17. reg 命令　构成连续 LQG 控制器。

格式：[Ac, Bc, Cc, Dc]= reg(A, B, C, D, K, L)

式中，K 为由 lqr 命令得到的反馈增益矩阵；L 为由 lqe 命令得到的 Kalman 滤波器增益矩阵。

结果：$\dot{\hat{x}} = (A-BK-LC+LDK)\hat{x} + Ly$

$\hat{u} = K\hat{x}$

18. dreg 命令　构成离散 LQG 控制器。

格式：[Ac, Bc, Cc, Dc]= dreg(A, B, C, D, K, L)

式中，K 为反馈增益矩阵；L 为 Kalman 滤波器增益矩阵。

结果：

$$\bar{x}(k+1) = [A-ALC-(B-ALD)E(K-KLC)]\bar{x}(k) + [AL-(B-ALD)EKL]y(k)$$

$$\hat{u}(k) = [K-KLC+KLDE(K-KLC)]\bar{x}(k) + (KL+KLDEKL)y(k)$$

式中，$E = (I+KLD)^{-1}$。

10.1.2 模型转换

转换模型时采用 c2d 、c2dm、c2dt、d2c、d2cm、ss2tf、ss2zp 、tf2ss 、tf2zp 、zp2tf 、zp2ss 等命令。

1. c2d 命令　将连续系统离散化。

格式：[F, G]= c2d(A, B, T)

式中，A 、B 为连续系统的系统矩阵和输入矩阵；F 、G 为离散化后的系统矩阵和输入矩阵；T 为采样周期。

2. c2dm 命令　将带离散化方法选项的连续系统离散化。

格式：[Ad, Bd, Cd, Dd]= c2dm(A, B, C, D,'method')

式中，'method'用来说明离散化的方法，具体如下：'zoh'为假设对输入信号加零阶保持器；'foh'为假设对输入信号加一阶保持器；'tustin'为双线性变换法；'prewrap'为带频率预歪扭的双线性变换法；'matched'为 SISO 系统的零极点匹配法。

或　　[numd, dend]= c2dm(num, den, 'method')

式中，numd 、dend 分别为传递函数 $G(z)$的分子、分母系数矢量。

3. c2dt 命令　将带输入纯滞后的连续系统离散化。

格式：[Ad, Bd, Cd, Dd]＝ c2dt(A, B, C, D, λ)

式中，λ 为输入时滞，即连续系统状态方程为

$$\dot{x}(t) = Ax(t)+Bu(t-\lambda)$$

$$y(t) = Cx(t)+Du(t)$$

4. d2c 命令　将离散系统连续化。

格式：[A, B]＝ d2c(F, G, T)

式中，T 为采样周期；A $=(1/T)\ln(F)$；B $=(F-I)^{-1}AG$。

5. d2cm 命令　将带连续化方法选项的离散系统连续化。

格式：[Ac, Bc, Cc, Dc]＝ d2cm(F, G, C, D, T,'method')

式中，T 为采样周期；method 为连续化方法，其定义与 c2dm 中的一样。

6. ss2tf 命令　将状态空间模型转换成传递函数模型。

格式：[num, den]＝ ss2tf(A, B, C, D, iu)

式中，iu 为输入的序号。转换的公式为

$$H(s)=\frac{\mathrm{num}(s)}{\mathrm{den}(s)}=C(sI-A)^{-1}B+D$$

7. ss2zp 命令　将状态空间模型转换成零极点增益模型。

格式：[Z, P, K]＝ ss2zp(A, B, C, D, iu)

式中，iu 为输入的序号。

8. tf2ss 命令　将传递函数模型转换成状态空间模型。

格式：[A, B, C, D]＝ tf2ss(num, den)

9. tf2zp 命令　将传递函数模型转换成零极点增益模型。

格式：[Z, P, K]＝ tf2zp(num, den)

10. zp2ss 命令　将零极点增益模型转换成状态空间模型。

格式：[A, B, C, D]＝ zp2ss(Z, P, K)

11. zp2tf 命令　将零极点增益模型转换成传递函数模型。

格式：[num, den]＝ zp2tf(Z, P, K)

10.1.3 模型降阶和最小实现

进行模型降阶和最小实现时可采用 balreal、dbalreal、modred、dmodred、minreal 等命令。

1. balreal 命令　求状态方程的均衡实现。

格式：[Ab, Bb, Cb, G, T]＝balreal(A, B, C)

式中， G 和 T 分别为均衡矩阵和变换矩阵。它通过变换矩阵对原系统进行均衡变换，以解决原系统的系统矩阵(A, B, C)中，因矩阵元素相差悬殊而造成数值不稳定的问题。

2. dbalreal 命令　求离散状态方程的均衡实现，为 balreal 命令的离散版本。

格式：[Ab, Bb, Cb, G, T]＝dbalreal(A, B, C)

3. modred 命令　将模型降阶。

格式：[Ar, Br, Cr, Dr]＝modred(A, B, C, D, elim)

式中， elim 为要消去的状态变量的序号。它常与 balreal 命令连用。

4. dmodred 命令　将离散模型降阶，为 modred 命令的离散版本。

格式：[Ar, Br, Cr, Dr]＝dmodred(A, B, C, D, elim)

5. minreal 命令　求状态方程的最小实现。

格式：[Am, Bm, Cm, Dm]＝minreal(A, B, C, D)

[Am, Bm, Cm, Dm]＝minreal(A, B, C, D, tol)

[Zm, Pm]＝minreal(Z, P)

[Zm, Pm]＝minreal(Z, P, tol)

[numm, denm]＝minreal(num, den)

[numm, denm]＝minreal(num, den, tol)

式中， tol 为允许误差，若省略则自动取为 eps 。

10.1.4　模型实现

模型实现时可采用 canon 、 ctrbf 、 ss2ss 等命令。

1. canon 命令　求系统标准型。

格式：[As, Bs, Cs, Ds, T]＝canon(A, B, C, D,'type')

式中， T 为变换矩阵；'type'为转换方法，它可有两种选择，即 'modal' 为对角标准型；'companion' 为伴随矩阵型。

2. ctrbf 命令 求能控标准型。

格式：[Ab, Bb, Cb, T, K]＝ ctrbf(A, B, C)

[Ab, Bb, Cb, T, K]＝ ctrbf(A, B, C, tol)

式中， tol 为允许误差； T 为转换矩阵； K 为各块的秩。若($\boldsymbol{A}$，$\boldsymbol{B}$)的能控判断矩阵的秩≤ $\boldsymbol{A}$ 的阶数 n 时，则存在变换矩阵 $\boldsymbol{T}$，使得

$$\boldsymbol{A}_{\mathrm{b}} = \boldsymbol{TAT}' \quad \boldsymbol{B}_{\mathrm{b}} = \boldsymbol{TB} \qquad \boldsymbol{C}_{\mathrm{b}} = \boldsymbol{CT}'$$

其中，$\boldsymbol{A}_{\mathrm{b}}$、$\boldsymbol{B}_{\mathrm{b}}$ 的形式为

$$\boldsymbol{A}_{\mathrm{b}} = \begin{bmatrix} \boldsymbol{A}_{\mathrm{nc}} & 0 \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{\mathrm{c}} \end{bmatrix} \qquad \boldsymbol{B}_{\mathrm{b}} = \begin{bmatrix} 0 \\ \boldsymbol{B}_{\mathrm{c}} \end{bmatrix} \qquad \boldsymbol{C}_{\mathrm{b}} = [\boldsymbol{C}_{\mathrm{nc}} \quad \boldsymbol{C}_{\mathrm{c}}]$$

式中，($\boldsymbol{A}_{\mathrm{c}}$, $\boldsymbol{B}_{\mathrm{c}}$)是能控的，且有 $\boldsymbol{C}_{\mathrm{c}}(s\boldsymbol{I}-\boldsymbol{A}_{\mathrm{c}})^{-1}\boldsymbol{B}_{\mathrm{c}} = \boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}$ 。$\boldsymbol{A}_{\mathrm{nc}}$ 是不能控的。即该命令将系统分解为能控和不能控的子空间。

3. obsvf 命令 求能观标准型。

格式：[Ab, Bb, Cb, T, K]＝ obsvf(A, B, C)

[Ab, Bb, Cb, T, K]＝ obsvf(A, B, C, tol)

式中， tol 为允许误差； T 为变换矩阵。若($\boldsymbol{A}$, $\mathbf{C}$)的能观判断矩阵的秩≤ $\boldsymbol{A}$ 的阶数 n 时，则存在变换矩阵 $\boldsymbol{T}$，使得

$$\boldsymbol{A}_{\mathrm{c}} = \boldsymbol{TAT}' \quad \boldsymbol{B}_{\mathrm{c}} = \boldsymbol{TB} \qquad \boldsymbol{C}_{\mathrm{b}} = \boldsymbol{CT}'$$

$\boldsymbol{A}_{\mathrm{c}}$、$\boldsymbol{B}_{\mathrm{c}}$ 的形式为

$$\boldsymbol{A}_{\mathrm{c}} = \begin{bmatrix} \boldsymbol{A}_{\mathrm{no}} & \boldsymbol{A}_{12} \\ 0 & \boldsymbol{A}_{\mathrm{o}} \end{bmatrix} \qquad \boldsymbol{B}_{\mathrm{c}} = \begin{bmatrix} \boldsymbol{B}_{\mathrm{no}} \\ \boldsymbol{B}_{\mathrm{o}} \end{bmatrix} \qquad \boldsymbol{C}_{\mathrm{c}} = \begin{bmatrix} 0 & \boldsymbol{C}_{\mathrm{o}} \end{bmatrix}$$

式中，($\boldsymbol{A}_{\mathrm{o}}$, $\boldsymbol{C}_{\mathrm{o}}$)是可观测的，$\boldsymbol{A}_{\mathrm{no}}$ 是不可观测的。

4. ss2ss 命令 求相似变换。

格式：[At, Bt, Ct, Dt]＝ ss2ss(A, B, C, D, T)

它执行相似变换 $\boldsymbol{z} = \boldsymbol{Tx}$，结果为

$$\dot{\boldsymbol{z}} = [\boldsymbol{TAT}^{-1}]\boldsymbol{z} + [\boldsymbol{TB}]\boldsymbol{u}$$

$$\boldsymbol{y} = [\boldsymbol{CT}^{-1}]\boldsymbol{z} + \boldsymbol{Du}$$

10.1.5 模型性质

求模型性质时可采用 covar 、 ctrb 、 damp 、 dcgain 、

dcovar、ddamp、ddcgain、dgram、dsort、esort、gram、obsv、printsys、tzero、tzero2 等命令。

1. covar 命令　求连续系统对白噪声的协方差响应。

格式：[P, Q]= covar(A, B, C, D, w)

　　　[P, Q]= covar(num, den, w)

式中，w 为高斯白噪声输入的强度 w，即 $E\{w(t)w'(\tau)\}= w\ \delta(t-\tau)$；P 和 Q 分别为系统的输出 $\boldsymbol{P}$ 和状态协方差响应 $\boldsymbol{Q}$，即 $\boldsymbol{P}= E\{\boldsymbol{y}\boldsymbol{y}'\}$，$\boldsymbol{Q}= E\{\boldsymbol{x}\boldsymbol{x}'\}$。

2. dcovar 命令　求离散系统对白噪声的协方差响应，为 covar 的离散版本。

格式：[P, Q]= dcovar(A, B, C, D, w)

　　　[P,Q]= dcovar(num, den, w)

3. damp 命令　求连续系统的自然频率和阻尼系数。

格式：[wn, Z]= damp(A)

式中，wn 为自然振荡频率ω_n；Z 为阻尼系数 ξ；A 可为下列几种情况之一：① A 为方阵，则认为它是系数矩阵；② A 为行矢量，则认为其是取自传递函数的多项式系数；③ A 为列矢量，则认为是包含系统的根。

4. ddamp 命令　求离散系统的自然频率和阻尼系数。

格式：[MAG, wn, Z]= ddamp(A, Ts)

式中，MAG 为 A 的 Z 平面幅值；wn 和 Z 分别为 A 的 s 平面等值自然振荡频率和阻尼系数；Ts 为采样周期。A 的格式与 damp 的要求相同。

5. dcgain 命令　求连续系统的稳态增益。

格式：K = dcgain(A, B, C, D)

　　　K = dcgain(num, den)

6. ddcgain 命令　求离散系统的稳态增益，为 dcgain 的离散版本。

格式：K = ddcgain(A, B, C, D)

K = ddcgain(num, den)

7. ctrb 命令　求能控性矩阵。

格式：Co = ctrb(A, B)

8. obsv 命令　求能观性矩阵。

格式：Ob = obsv(A, C)

9. esort 命令　对连续系统复数特征值排序。

格式：S = esort(P)

它对 P 中的复数特征值，按实部的大小递减排序。

10. dsort 命令　对离散系统复数的特征值排序，为 esort 的离散版本。

格式：S = dsort(P)

11. gram 命令　求连续系统的能控性和能观性检测矩阵。

格式：Gc = gram(A, B)

Go = gram(A′, C′)

式中，
$$Gc = \int_0^\infty e^{tA} \boldsymbol{BB}' e^{tA'} dt$$

$$Go = \int_0^\infty e^{tA'} \boldsymbol{CC}' e^{tA} dt$$

当且仅当($\boldsymbol{A}$，$\boldsymbol{B}$)能控时，$\boldsymbol{G}_c$是满秩的；当且仅当($\boldsymbol{A}$, $\boldsymbol{C}$)能观时，$\boldsymbol{Go}$ 是满秩的。

12. dgram 命令　求离散系统的能控性和能观性检测矩阵，为 gram 的离散版本。

格式：Gc = dgram(A, B)

Go = dgram(A′, C′)

13. tzero 命令　求传输零点。

格式：Z = tzero(A, B, C, D)

14. tzero2 命令　求传输零点。

格式：Z = tzero2(A, B, C, D)

15. printsys 命令　打印出系统状态方程或传递函数。

格式： printsys(A, B, C, D, ulabels, ylabels, xlabels)

printsys(num, den,'s')

printsys(num, den,'z')

10.1.6 时域响应

求时域响应可采用 dimpulse 、 dinitial 、 dlsim 、 dstep 、 filter 、 impulse 、 initial 、 lsim 、 ltitr 、 step 、 stepfun 等命令。部分时域响应命令具有自动屏幕绘图功能，即在使用时，若无等式左边的量，也就是用命令本身，则会进行自动屏幕绘图；若有等式左边的量，则不进行屏幕绘图；具有这种功能的命令有 impulse 、 dimpulse 、 step 、 dstep 、 lsim 、 dlsim 、 initial 、 dinitial 。另外，连续时域响应命令一般还具有自动时间矢量选择功能。若在命令的输入变量部分给出了时间矢量 t，则按给定的时间矢量进行系统仿真；若命令的输入变量部分未给出时间矢量，命令会根据系统的状态方程或传递函数、初始条件、精度要求等自动选择出时间矢量，并在等式左边的结果要求给出 t 时，返回 t 矢量。对离散时域响应命令来说，自动时间矢量选择功能就变成自动采用周期个数选择功能。

1. impulse 命令 求脉冲响应。

格式：[y, x]= impulse(A, B, C, D, iu, t)

[y, x]= impulse(num, den, t)

[y, x, t']= impulse(A, B, C, D, iu)

[y, x, t]= impulse(num, den)

2. dimpulse 命令 求离散系统单位脉冲响应。

格式：[y, x]= dimpulse(A, B, C, D, iu, n)

[y, x]= dimpulse(num, den, n)

[y, x, n]= dimpulse(A, B, C, D, iu)

[y, x, n]= dimpulse(num, den)

式中， n 为采样点数。

3. step 命令 求阶跃响应。

格式：[y, x]= step(A, B, C, D, iu, t)

[y, x]= step(num, den, t)

[y, x, t]= step(A, B, C, D, iu)

[y, x, t]= step(num, den)

4. dstep 命令 求离散系统阶跃响应。

格式：[y, x]= dstep(A, B, C, D, iu, n)

[y, x]= dstep(num, den, n)

[y, x, n]= dstep(A, B, C, D, iu)

[y, x, n]= dstep(num, den)

5. lsim 命令 对任意输入的连续系统进行仿真。

格式：[y, x]= lsim(A, B, C, D, u, t)

[y, x]= lsim(A, B, C, D, u, t, x0)

[y, x]= lsim(num, den, u, t)

[y, x, t]= lsim(A, B, C, D, u)

[y, x, t]= lsim(A, B, C, D, u, x0)

[y, x, t]= lsim(num, den, u)

式中， u 为输入矢量； x0 为初始条件。

6. dlsim 命令 对任意输入的离散系统进行仿真。

格式：[y, x]= dlsim(A, B, C, D, u)

[y, x]= dlsim(A, B, C, D, u, x0)

[y, x]= dlsim(num, den, u)

7. initial 命令 求带初始条件的连续系统时域响应。

格式：[y, x]= initial(A, B, C, D, x0, t)

[y, x, t]= initial(A, B, C, D, x0)

8. dinitial 命令 求带初始条件的离散系统时域响应。

格式：[y, x]= dinitial(A, B, C, D, x0, n)

[y, x, n]= dinitial(A, B, C, D, x0)

9. filter 命令 对 SISO 系统的 z 变换进行仿真。

本命令对矢量 x 进行滤波，滤波器由 $\boldsymbol{B}$ 、 $\boldsymbol{A}$ 给出。

所用公式为

$$y(n)=b(1)x(n)+\cdots+b(n_b+1)x(n-n_b)-a(2)y(n-1)$$
$$-\cdots-a(n_a+1)y(n-n_a)$$

10. ltitr 命令 求线性定常系统时域响应。

格式： x = ltitr(A, B, u)

x = ltitr(A, B, u, x0)

11. stepfun 命令 求阶跃函数。

格式： t1 = stepfun(t, t0)

式中， t 为一单调增大的矢量 t； t1 为矢量元素 t_1，格式与 t 同，且有

$$t_1=\begin{cases}0 & t<t_0\\ 1 & t>t_0 \quad \& \quad t=t_0\end{cases}$$

10.1.7 频域响应

求频域响应时可采用bode、dbode、dnichols 、 dnyquist 、 dsigma 、 fbode 、 freqs 、 freqz 、 ltifr 、 margin 、 nichols 、 ngrid 、 nyquist 、 sigma 等命令。频域响应命令一般具有自动屏幕绘图功能，即在使用时，若无等式左边的量，也就是用命令本身，则会进行自动屏幕绘图；若有等式左边的量，则不进行自动屏幕绘图。另外，频域响应命令一般还具有自动频率矢量选择功能。若在命令输入变量部分给出了频率矢量ω，则按给定的频率矢量进行频率响应分析。若命令的输入变量部分未给出频率矢量，则命令会根据系统的状态方程或传递函数等自动选择出频率矢量，并在等式左边要求给出ω 时，返回矢量ω 。

1. bode 命令 绘制伯德图。

格式：[M, P]= bode(A, B, C, D, iu, w)

[M, P]= bode(num, den, w)

[M, P, w]= bode(A, B, C, D, iu)

[M, P, w]= bode(num, den)

对状态空间表示有

$$G(s)=\boldsymbol{C}(s\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}$$

M(w)＝ abs(G(jw))

P(w)＝ angle(G(jw))

M 可用 20log10(M)转换成以 dB 为单位， w 可用 logspace 命令进行对数型分段。

2. dbode 命令　绘制离散系统伯德图。

格式：[M, P]＝ dbode(A, B, C, D, Ts , iu, w)

[M, P]＝ dbode(num, den, Ts, w)

[M, P, w]＝ dbode(A, B, C, D, Ts, iu)

[M, P, w]＝ dbode(num, den, Ts)

式中， Ts 为采样周期 T_s。

3. fbode 命令　绘制快速伯德图。

其格式与 bode 命令完全相同。与 bode 命令的区别是计算比 bode 命令快，但精度比 bode 命令的差。

4. nyquist 命令　绘制奈魁斯特图。

格式：[Re, Im]＝ nyquist(A, B, C, D, iu, w)

[Re, Im]＝ nyquidt(num, den, w)

[Re, Im, w]＝ nyquist(A, B, C, D, iu)

[Re, Im, w]＝ nuquist(num,　den)

5. dnyquist 命令　绘制离散系统奈魁斯特图。

格式：[Re, Im]＝ dnyquist(A, B, C, D, Ts, iu, w]

[Re, Im]＝ dnyquist(num, den, Ts,u]

[Re, Im, w]＝ dnyquist(A, B, C, D, Ts,iu]

[Re, Im, w]＝ dnyquist(num, den, Ts)

式中， Ts 为采样周期。对离散状态方程($\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$)，有：

$$G(\omega)=\boldsymbol{C}(e^{j\omega T}\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}$$

Re(w)＝ real(G(w))

Im(w)＝ imag(G(w))

6. nichols 命令　绘制尼科尔斯图。

格式：[M, P]= nichols(A, B, C, D, iu, w]

[M, P]= nichols(num, den, w]

[M, P, w]= nichols(A, B, C, D, iu)

[M, P, w]= nichols(num, den)

7. dnichols 命令 绘制离散系统尼科尔斯图。

格式：[M, P]= dnichols(A, B, C, D, Ts,iu, w)

[M, P]= dnichols(num, den, w)

[M, P, w]= dnichols(A, B, C, D, Ts, iu)

[M, P, w]= dnichols(num, den)

8. ngrid 命令 为尼科尔斯图绘制格线。

格式： ngrid

ngrid('new')

格线的范围幅值为–40dB 到+40dB，相角为–360°到 0°。格距 $H/(1+H)$，H 为任一复数。ngrid 对现存的尼科尔斯图画格线。若用 ngrid('new')，则先清图，再画格线。

9. sigma 命令 求线性连续系统的奇异值频率响应。

格式： sv = sigma(A, B, C, D, w)

sv = sigma(A, B, C, D, w, 'inv')

[sv, w]= sigma(A, B, C, D)

[sv, w]= sigma(A, B, C, D, 'inv')

奇异值是伯德图中幅频特性在 MIMO 系统中的扩展。MIMO 系统对应频率特性矩阵可写成

$$\boldsymbol{G}(\mathrm{j}\omega)= \boldsymbol{C}(\mathrm{j}\omega \boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}$$

而其逆矩阵为

$$\boldsymbol{G}^{-1}(\mathrm{j}\omega)=[\boldsymbol{C}(\mathrm{j}\omega \boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{B}+\boldsymbol{D}]^{-1}$$

当输入项包含 'inv' 时，计算的是 $\boldsymbol{G}^{-1}(\mathrm{j}\omega)$的频率特性；无 'inv' 时，计算的是 $\boldsymbol{G}(\mathrm{j}\omega)$的频率特性。

10. dsigma 命令 求线性离散系统的奇异值频率响应。

格式： sv = dsigma(A, B, C, D, Ts, w)

sv = dsigma(A, B, C, D, Ts, w, 'inv')

[sv, w] = dsigma(A, B, C, D, Ts)

[sv, w] = dsigma(A, B, C, D, Ts, 'inv')

11. freqs 命令 求拉氏变换频率响应。

格式： H = freqs(B, A, w)

它计算由$H(s) = B(s)/A(s)$表示的传递函数的频率特性。$H(s)$的分子、分母的各系数分别由矢量B和A降幂给出。由矢量ω给出的频率点处算出的复数频率特性存放在H中。

12. freqz 命令 求z变换数字滤波器频率响应。

格式：[H, w] = freqz(B, A, N)

[H, w] = freqz(B, A, N, 'whole')

H = freqz(B, A, w)

式中， N 为计算的频率点数，对应的 w 值为单位圆内上半区均匀分布的频率点。计算出来的$H(e^{j\omega T})$存放在 H 中。若在输入项中包含'whole'，则 w 分布在整个单位圆中。若输入项中包含 w，则按给出的 w 进行计算。

它计算由$H(z) = B(z)/A(z)$表示的传递函数的频率特性。$H(z)$由下式给出：

$$H(z) = \frac{B(z)}{A(z)} = \frac{b(1) + b(2)z^{-1} + \cdots + b(nb+1)z^{-nb}}{1 + a(2)z^{-1} + \cdots + a(na+1)z^{-na}}$$

13. ltifr 命令 求线性定常系统的频率响应。

格式： G = ltifr(A, b, s)

式中， s 为给定的复数频率矢量。它计算$G(s) = (sI - A)^{-1}\boldsymbol{b}$。

14. margin 命令 求幅值和相角裕量及幅值和相位交界频率。

格式：[GM, PM, wcg, wcp] = margin(M, P, w)

式中， M 、 P 分别对应频率矢量ω的幅频特性值和相频特性值矢量； GM 为增益裕量； PM 为相角裕量； wcg 为增益交界频率； wcp 为相位交界频率。

10.1.8 根轨迹

求根轨迹时可采用pzmap、rlocfind、rlocus、sgrid、zgrid等命令。

1. pzmap 命令 绘制线性连续系统的零极点图。

格式：[p, z]= pzmap(A, B, C, D)

[p, z]= pzmap(num, den)

式中，p 和 z 分别为系统(A, B, C, D)或 num(s)/den(s)的极点矢量和传输零点矢量（或矩阵）。若无等式右边项，则在 s 平面上画出对应极、零点位置。极点用“×”表示，零点用“ o ”表示。

2. rlocus 命令 绘制根轨迹图。

格式：R = rlocus(num, den, k)

式中，k 为增益变化矢量。它计算 1+k*num(s)/den(s)= 0 的根。结果 R 为 length(k)行、length(den)−1 列。若无等式左边的 R 时，则在屏幕上画出根轨迹。

3. rlocfind 命令 找出给定的一组根对应的根轨迹增益。

格式：[K, P]= rlocfind(A, B, C, D)

[K, P]= rlocfind(num, den)

它要求在屏幕上先已绘制好有关根轨迹图。然后，此命令将产生一个光标以用来选择希望的闭环极点。命令执行结果：K 为对应选择点处的根轨迹增益；P 为此点处的系统闭环特征根。

4. sgrid 命令 在 s 平面上绘制连续系统根轨迹的ω_n、Z的格线。

格式：sgrid

sgrid('new')

sgrid(Z, wn)

sgrid 在现存的屏幕根轨迹或极零点图上画出自然振荡频率ω_n、阻尼比矢量 Z 对应的格线。sgrid('new')是先清屏再画格线。而 sgrid(Z, wn)则绘制阻尼比矢量 Z 、自然振荡频率矢量ω_n的格线。

5. zgrid 命令　在 z 平面上绘制离散系统根轨迹的ω_n、Z的格线。

格式：zgrid('new')

zgrid(Z, wn)

10.1.9 增益选择

选择增益时采用acker、dlqe、dlqew、dlqr、dlqry、lqe、lqe2、lqed、lqew、lqr、lqrd、lqry、lqr2、place等命令，这些主要是二次型设计和极点配置方面的命令。

1. acker 命令　求 SISO 系统的极点配置。

格式：K = acker(A, b, P)

对单输入系统$\dot{\boldsymbol{x}} = \boldsymbol{Ax}+\boldsymbol{bu}$，该命令求取状态反馈增益矩阵$\boldsymbol{K}$，及状态反馈$\boldsymbol{u} = -\boldsymbol{Kx}$，使得闭环系统极点为给定极点$\boldsymbol{P}$，即$\boldsymbol{P}$ = eig($\boldsymbol{A}$−$\boldsymbol{b}$*$\boldsymbol{K}$)。

2. place 命令　求极点配置。

格式：K = place(A, B, P)

它用于 MIMO 系统。

3. lqr 命令　设计线性二次型调节器。

格式：K = lqr(A, B, Q, R, N)

[K, P, E]= lqr(A, B, Q, R, N)

式中，N 为交叉乘积项$\boldsymbol{N}$，它是可选项；P 为 Riccati 方程

$$\boldsymbol{A}'\boldsymbol{P}+\boldsymbol{PA}-(\boldsymbol{PB}+\boldsymbol{N})\boldsymbol{R}^{-1}(\boldsymbol{PB}+\boldsymbol{N})'+\boldsymbol{Q} = 0$$

的解；E 为$\boldsymbol{A}-\boldsymbol{BK}$的特征值。

它计算最优反馈增益矩阵$\boldsymbol{K}$，使得反馈控制率$\boldsymbol{u} = -\boldsymbol{Kx}$，最小化如下性能指标

$$J = \int_0^{\infty} (\boldsymbol{x}'\boldsymbol{Qx}+\boldsymbol{u}'\boldsymbol{Ru}+2\boldsymbol{x}'\boldsymbol{Nu})\mathrm{d}t$$

4. lqr2 命令　用 Schur 方法设计线性二次型调节器。

格式：K = lqr2(A, B, Q, R, N)

[K, P, E]= lqr2(A, B, Q, R, N]

5. lqry 命令　设计线性二次型输出调节器。

格式：[K, P]= lqry(A, B, C, D, Q, R)

它计算最优反馈增益矩阵 $\boldsymbol{K}$，使得反馈控制率 $\boldsymbol{u}=-\boldsymbol{K}\boldsymbol{y}$,其最小化性能指标为 $J=\int_0^\infty(\boldsymbol{y}'\boldsymbol{Q}\,\boldsymbol{y}+\boldsymbol{u}'\boldsymbol{R}\,\boldsymbol{u})\,\mathrm{d}t$； P 为 Riccati 方程 $\boldsymbol{A}'\boldsymbol{P}+\boldsymbol{P}\boldsymbol{A}-\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{P}\boldsymbol{B}'+\boldsymbol{Q}=0$ 的解。

6. dlqr 命令　设计离散线性二次型调节器。

格式：[K, P, E]= dlqr(A, B, Q, R, N)

式中， K 为离散最优反馈增益矩阵；目标函数为

$$J=\sum_{k=0}^{\infty}[\boldsymbol{x}'(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}'(k)\boldsymbol{R}\,\boldsymbol{u}(k)+2\boldsymbol{x}'(k)\boldsymbol{N}\,\boldsymbol{u}(k)]$$

P 为离散 Riccati 方程的解； E 为 $\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K}$ 的特征值。

7. dlqry 命令　设计离散线性二次型输出调节器，为 lqry 的离散版本。

格式：[K, P, E]= dlqry(A, B, C, D, Q, R)

8. lqrd 命令　用连续系统的性能指标函数来进行离散二次型调节器设计。

格式：[K, P, E]= lqrd(A, B, Q, R, Ts)

式中， A 、 B 为连续系统状态方程的系数矩阵； Q 、 R 为连续系统目标函数的加权矩阵；Ts 为采样周期。此命令先对 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{Q}$ 、 $\boldsymbol{R}$ 以 T_s 及零阶保持器离散化，再进行离散二次型调节器设计。

9. lqe 命令　求 Kalman 滤波器的参数。

格式：[L, P, E]= lqe(A, F, C, Q0, R0, N0)

式中， L 为 Kalman 滤波器的增益矩阵； P 为对应 Riccati 方程的解； E 为闭环特征值。系统状态方程为

$$\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}+\boldsymbol{F}\boldsymbol{w}$$
$$\boldsymbol{y}=\boldsymbol{C}\boldsymbol{x}+\boldsymbol{D}\boldsymbol{u}+\boldsymbol{v}$$

过程噪声参数为 $E\{\boldsymbol{w}\}=E\{\boldsymbol{v}\}=0$， $E\{\boldsymbol{w}\,\boldsymbol{w}'\}=\boldsymbol{Q}_0$， $E\{\boldsymbol{v}\,\boldsymbol{v}'\}=$

R_0， $E\{w v'\} = N_0$。

10. lqe2 命令 用 Schur 方法求 Kalman 滤波器的参数。

格式：[L, P, E]= lqe2(A, F, C, Q0, R0, N0)

11. lqew 命令 设计通用 Kalman 滤波器。

格式：[L, P, E]= lqew(A, F, C, P0, Q0, R0, N0)

与 lqe 相比，系统输出方程变为 $\boldsymbol{y} = \boldsymbol{Cx}+\boldsymbol{Du}+\boldsymbol{P_0w}+\boldsymbol{v}$ 。

12. lqed 命令 用连续系统的性能指标函数设计离散的 Kalman 滤波器。

格式：[L, M, P, E]= lqed(A, F, C, Q0, R0, Ts)

13. dlqe 命令 设计离散 Kalman 滤波器。

格式：[L, M, P, E]= dlqe(A, F, C, Q0, R0)

式中， L 为 Kalman 滤波器增益矩阵； M 为 Riccati 方程的解； P 为测量更新后的估计误差协方差 $P = E\{[\hat{x}-x][\hat{x}-x]'\}$； E 为 Kalman 滤波器的闭环特征值 $E = \mathrm{eig}(A-A*L*C)$

14. dlqew 命令 通用离散 Kalman 滤波器设计。

格式：[L, M, P, E]= dlqew(A, F, C, P0, Q0, R0)

10.1.10 方程求解

求解方程时可采用 are 、 dlyap 、 lyap 、 lyap2 等命令。

1. are 命令 求解 Riccati 代数方程。

格式： X = are(A, B, C)

式中， X 为 Riccati 方程 $A'X + XA - XBX + C = 0$ 的解。

2. lyap 命令 求解连续系统 Liapunov 方程。

格式： X = lyap(A, C)

X = lyap(A, B, C)

X = lyap(A, C)求解特殊型 Liapunov 方程 $AX + XA' =-C$ ；X = lyap(A, B, C)求解一般型 Liapunov 方程 $AX + XB =-C$ 。

3. lyap2 命令 用特征值分解方法求解 Liapunov 方程。

格式： X = lyap2(A, C)

X = lyap2(A, B, C)

4. dlyap 命令　求解离散系统的 Liapunov 方程。

格式：X = dlyap(A, C)

它用于求解 Liapunov 方程 $\boldsymbol{AXA}' + \boldsymbol{C} = \boldsymbol{X}$。

10.1.11 演示程序

控制系统工具箱提供了五个演示程序，用以帮助理解命令的使用。

1. Ctrldemo 控制系统工具箱一般功能介绍。

2. boildemo 锅炉系统的 LQG 设计。

3. jetdemo 喷气飞机交通偏航阻尼器设计。

4. diskdemo 硬盘控制器的数字控制设计。

5. Kalmdemo　Kalman 滤波器设计与仿真。

10.2 系统辨识工具箱

系统辨识就是从观测到的含有噪声的系统输入输出数据中提取所研究系统的数学模型。这种数学模型实际上是对系统输入输出特性在某种准则意义下的一种近似，近似的程度取决于对系统的了解及所选用的辨识方法是否合理。

设系统的输入为 $\boldsymbol{u}(k)$ $(k = 1, 2, \cdots, N)$；输出为 $\boldsymbol{y}(k)$ $(k = 1, 2, \cdots, N)$。用差分方程表示则为

$$\boldsymbol{y}(k) = \boldsymbol{G}(z)\boldsymbol{u}(k) + \boldsymbol{v}(k) \tag{10-6}$$

式中，$\boldsymbol{v}(k)$为系统噪声，它可以表示为

$$\boldsymbol{v}(k) = \boldsymbol{H}(z)\boldsymbol{e}(k) \tag{10-7}$$

式中，$\boldsymbol{e}(k)$为方差是 λ 的白噪声。

综合式(10-6)、(10-7)有：

$$\boldsymbol{y}(k) = \boldsymbol{G}(z)\boldsymbol{u}(k) + \boldsymbol{H}(z)\boldsymbol{e}(k) \tag{10-8}$$

为了估计式(10-8)中的 $\boldsymbol{G}(z)$和 $\boldsymbol{H}(z)$，要将其转换成辨识用的参数模型。

通用的参数模型如下：

$$A(z)y(k) = \frac{B(z)}{F(z)}u(k-n_k) + \frac{C(z)}{D(z)}e(k) \tag{10-9}$$

式中，$A(z)$、$B(z)$、$C(z)$、$D(z)$、$F(z)$均为 z^{-1} 的多项式， 即：

$$
\begin{aligned}
A(z) &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\
B(z) &= b_1 + b_2 z^{-1} + \cdots + a_{n_b} z^{-n_b+1} \\
C(z) &= 1 + c_1 z^{-1} + \cdots + a_{n_c} z^{-n_c} \\
D(z) &= 1 + d_1 z^{-1} + \cdots + a_{n_d} z^{-n_d} \\
F(z) &= 1 + f_1 z^{-1} + \cdots + a_{n_f} z^{-n_f}
\end{aligned}
\qquad (10-10)
$$

一些常用的参数模型都是通用数学模型的特殊情况。如 ARX 模型为

$$A(z)y(k) = B(z)u(k-n_k)+e(k) \qquad (10\text{-}11)$$

对应通用参数模型中的 $n_c = n_d = n_f = 0$， ARMAX 模型为

$$A(z)y(k) = B(z)u(k-n_k)+C(z)e(k) \qquad (10\text{-}12)$$

对应通用参数模型中的 $n_c = n_f = 0$，输出误差(OE)模型为

$$y(k) = \frac{B(z)}{F(z)}u(k-n_k)+e(k) \qquad (10\text{-}13)$$

对应通用参数模型中的 $n_a = n_c = n_d = 0$； Box-Jenkins 模型为

$$y(k) = \frac{B(z)}{F(z)}u(k-n_k)+\frac{C(z)}{D(z)}e(k) \qquad (10\text{-}14)$$

对应通用参数模型中的 $n_a = 0$ 。

在系统辨识工具箱中使用的参数模型的基本格式为 theta 格式；其与式(10-9)的通用参数模型相关。在 theta 矩阵中包含模型结构（即 $n_a, n_b, n_c, n_d, n_f, n_k$ 等结构参数）、估计参数（即 A, B, C, D, F 等多项式的参数）和估计精度等。

除了 theta 格式以外，还有 Freqfunc （频率函数）格式，为 N+1 行、2 ～ 5 列的矩阵，第一行为整数，用以指示该列的情况。通常的各列安排为频率值、幅值、幅值标准差、相位值、相位标准差；多项式格式为 A 、B 、C 、D 、F 以降幂的行矢量方式给出，如 A =[1 a_1 … a_n]；零极点格式为各列排列的顺序是 G 的零点、G 的极点、H 的零点、H 的极点。

以上这四种模型格式可由有关建立模型的命令得到，并可通

过一些命令进行相互转换。

系统辨识工具箱就其功能来说，可以分成以下几个部分：

（1）参数估计；

（2）非参数估计；

（3）数据处理；

（4）模型结构定义；

（5）模型转换；

（6）递推参数估计；

（7）模型结构处理；

（8）模型表达；

（9）信息提取；

（10）模型结构选择；

（11）模型不确定性评估和模型校验。

下面分别简要介绍各部分主要命令的功能。

10.2.1 参数估计

进行参数估计时可采用 ar 、 armax 、 arx 、 bj 、 canstart 、iv 、 ivar 、 iv4 、 ivx 、 oe 、 pem 等命令。

1. ar 命令 求 AR 模型的参数估计。

格式： th = ar(y, n)

th = ar(y, n,'method')

th = ar(y, n,'win')

式中，'method' 为辨识方法，有'fb'、'ls'、'yw'、'burg'、'gl'等几种方法可以选择。'win'有'now'、'prw'、'ppw'几种数据窗可供选择。

2. armax 命令 求 ARMAX 模型的预测误差估计。

格式： th = armax(z, nn)

式中，z =[y u]，nn =[na nb nk]或 z = y，nn =[na nc]。

模型为 $A(z)y(k)= B(z)u(k-n_k)+C(z)e(k)$

3. arx 命令 求 ARX 模型的最小二乘估计。

格式： th = arx(z, nn)

式中， z =[y u]， nn =[nb nc nd nf nk]。

模型为 $A(z)y(k)= B(z)u(k-nk)+e(k)$ 。

4. bj 命令 求 Box-Jenkins 模型的预测误差估计。

格式： th = bj(z, nn)

式中， z =[y u]， nn =[nb nc nd nf nk]。

模型： $y(k)=[B(z)/F(z)]u(k-nk)+[C(z)/D(z)]e(k)$

5. canstart 命令 求带初始参数估计的多变量 theta 模型。

格式： th1 = canstart(z, psobs, nu, oe)

式中， z =[y u], 每个输入/输出元素占一列； psobs 为用来定义多变量状态空间结构的伪可观测矩阵。 nu 为输入个数， oe 为输出误差选择。这里 th1 为系统的初始估计，应该用 th = pem(z, th1)进一步地改进。

6. iv 命令 求单输出 ARX 模型的辅助变量估计。

格式： th = iv(z, nn, NF, MF)

式中， z =[y u]， nn =[na nb nk]。

模型： $A(z)y(k)= B(z)u(k-n_k)+e(k)$

辅助变量： $NF(z)x(k)= MF(z)u(k)$

7. ivar 命令 求标量时间序列的 AR 部分的辅助变量估计。

格式： th = ivar(y, na)

th = ivar(y, na, nc)

8. iv4 命令 求 ARX 模型的近似最优辅助变量估计。

格式： th = iv4(z, nn)

式中， z =[y u]， nn =[na nb nk]。

9. ivx 命令 求 ARX 模型的辅助变量估计。

格式： th = ivx(z, nn, x)

式中， z =[y u]， nn =[na nb nk]， x 为辅助变量。

10. oe 命令 求输出误差模型的预测误差估计。

格式： th = oe(z, nn)

式中，z =[y　u]，nn =[nb　nf　nk]。

模型：$y(k)=[B(z)/F(z)]u(k-n_k)+e(k)$

11. pem 命令　求通用参数模型的预测误差估计。

格式：th = pem(z, thstruc)

式中，z =[y　u],thstruc 为定义模型结构的矩阵（例如：thstruc =[na　nb　nc　nd　nf　nk]）。

10.2.2　非参数估计

进行非参数估计时可采用 covf、cra、etfe、spa 等命令。

1. covf 命令　求数据矩阵的协方差函数估计。

格式：r = covf(z, m)

式中，z 为数据矩阵，典型的为 z =[y　u]，m 为最大的纯滞后 -1，r 为 z 的协方差矩阵。

2. cra 命令　求互相关分析。

格式：IR = cra(z)

式中，z =[y　u]，IR 为估计的脉冲响应。

3. etfe 命令　求实验传递函数和周期图。

格式：G = etfe(z)

式中，z =[y　u]，G 为标准频率函数格式。

4. spa 命令　频谱分析。

格式：G = spa(z)

式中，z =[y　u]，G 为标准频率函数格式。

10.2.3　数据处理

进行数据处理时可采用 dtrend、idfilt 等命令。

1. dtrend 命令　将数据组中的趋势值去掉。

格式：zd = dtrend(z)

　　　zd = dtrend(z, o, bp)

式中，z 为数据组；o = 0 时，去除样本的均值；o = 1 时，去除线性趋势值；bp 为分段去除线性趋势值的分段点。

2. idfilt 命令　用 Buttorworth 滤波器对数据滤波。

格式： zf = idfilt(z, n, wn)

式中， z =[y u];n 为滤波器的阶数。wn 为剪切频率。若以 wn 为标准，则用一个低通滤波器；若 wn =[w wh]，则用带通滤波器。

10.2.4 模型结构定义

定义模型结构时可采用 arx2th 、 canform 、 mf2th 、 modstruc 、 ms2th 、 poly2th 等命令。

1. arx2th 命令 求 ARX 模型的 theta 格式。

格式： th = arx2th(A, B, ny, nu)

式中， A 、 B 为 ARX 模型多项式， ny 和 nu 分别为输出和输入的维数。

2. canform 命令 定义正则模型结构。

格式： MS = canform(psd, nu, x0)

式中， psd 为伪能观性矩阵， nu 为输入维数， x0 为初始状态。

3. mf2th 命令 将用户定义的模型结构转换成 theta 格式。

格式： th = mf2th(model, cd, parval, aux, lambda, T)

式中， model 为用户所写的 m 文件名所指定的模型结构。该 m 文件应具有如下格式：

[a, b, c, d, k, x0]=用户文件名（参数, T, aux ）

式中， model 为‘用户文件名’； cd 为若‘用户文件名’提供的是一个调用时， T 值为负值的连续模型，则 cd ='c'，否则 cd ='d'； parval 为自由参数的值； aux 为上述用户所写 m 文件中的 aux 辅助参数； lambda 为新结构的协方差矩阵； T 为采样周期。缺省值为 aux =[ ];T = 1 ； lambda =单位矩阵。

4. modstruc 命令 构造用于 ms2th 命令的模型结构。

格式： MS = modstruc(A, B, C, D, K, x0)

式中， A 、 B 、 C 、 D 、 K 、 x0 为下列状态空间模型的矩阵：

$x_{new} = \boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)+\boldsymbol{K}\boldsymbol{e}(t)$

$\boldsymbol{y}(t)= \boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)+\boldsymbol{e}(t)$

其中，x_{new}为$x(t+T)$或者$dx(t)/dt$。x0为初始状态，在这些矩阵中，确定的参数以其参数值送入，而要估计的参数则在相应的位置上放上NaN。

5. ms2th 命令　将标准的状态空间参数形式转换成theta格式。

格式：THETA = ms2th(MS, cd, parval, lambda, T)

式中，MS为模型结构，指出了哪些矩阵参数是确定的，哪些参数是待估计的(对应位置为NaN)，此模型结构一般是由modstruc命令或canform命令产生的。cd用以区分连续与离散模型，cd ='c' 时为连续模型；cd ='d' 时为离散模型（缺省时为'd'）。parval为自由参数的值（缺省时为0）。lambda为新结构的协方差矩阵；T为采样间隔。

6. poly2th 命令　由给定多项式构成theta格式矩阵。

格式：th =[poly2th(A, B, C, D, F, lam, T)

式中，A、B、C、D、F为通用参数格式的多项式，A、C、D、F的首项为1，而B中允许有前导0，以表示滞后参数n_k；lam为噪声$e(k)$的方差；T为采样周期。

10.2.5 模型转换

进行模型转换时可采用th2arx、th2ff、th2par、th2poly、th2ss、th2tf、th2zp、thc2thd、thd2thc等命令。

1. th2arx 命令　将theta格式模型转换成ARX模型。

格式：[A， B]= th2arx(th)

2. th2ff 命令　计算模型的频率函数以及标准差。

格式：G = th2ff(th)

[G, NSP]= th2ff(th)

式中，G为频率函数，NSP为噪声频谱。

3. th2par 命令　将theta格式转换成参数和协方差矩阵。

格式：[par, p, lam]= th2par(th)

式中，par为参数矢量，p为估计参数的协方差矩阵，lam为新

结构的方差。

4. th2poly 命令　将 theta 格式转换成相关多项式。

格式：[A, B, C, D, F, lam]＝ th2poly(th)

式中，A、B、C、D、F 分别为对应通用参数模型中的多项式；lam 为噪声的方差。

5. th2ss 命令　将 theta 格式转换成状态空间模型。

格式：[A, B, C, D, K, x0]＝ th2ss(th)

式中，x0 为初始矢量；A、B、C、D、K 对应状态空间模型矩阵：

$$x_n = Ax + Bu + Ke$$

$$y = Cx + Du + e$$

式中，x_n 为 dx/dt（连续）或 $x(k+1)$（离散）。

6. th2tf 命令　将 theta 格式转换成传递函数。

格式：[num, den]＝ th2tf(th, iu)

式中，iu 为输入序号（省缺时为 1）。

7. th2zp 命令　计算模型的零极点，稳态增益及标准差。

格式：[zepo, k]＝ th2zp(th)

式中，zepo 为模型的零极点及标准差，k 为稳态增益。

8. thc2thd 命令　将连续模型转换成离散模型。

格式：thd ＝ thc2thd(thc, T)

式中，T 为采样周期。

9. thd2thc 命令　将离散模型转换成连续模型。

格式：thc ＝ thd2thc(th)

10.2.6　递推参数估计

进行递推参数估计时可采用 rarx、rarmax、rbj、roe、rpem、rplr、segment 等命令。

1. rarx 命令　求 ARX 模型的递推估计。

格式：[thm, yhat]＝ rarx(z, nn, adm, adg)

式中，z ＝[y　u]；nn ＝[na　nb　nk]；adm 为算法结构，adg

为算法增益。

若 adm ='ff'， adg = lam，则为遗忘因子是 lam 的遗忘因子法。

若 adm ='kf'， adg = R_1，则为每时间步参数变化的协方差为 R_1 的 Kalman 滤波器算法。

若 adm ='ng'， adg = gam，则为增益是 gam 的标准化的梯度算法。

若 adm ='ug'， adg = gam，则为增益为 gam 的非标准化梯度算法。

thm 为估计的结果； yhat 为预测的输出。参数初值（ th0 ）及 P 矩阵初始值（ p0 ）由命令[thm, yhat, p]= rarx(z, nn, adm, adg, th0, p0)给出。

2. rarmax 命令　求 ARMAX 模型的递推估计。

格式：　[thm, yhat]= rarmax(z, nn, adm, adg)

式中， z =[y　u]； nn =[na　nb　nc　nk]；其它参数可参考 rarx 。

3. rbj 命令　求 Box-Jenkind 模型的递推估计。

格式：[thm, yhat]= rbj(z, nn, adm, adg)

式中， z =[y　u]； nn =[nb　nc　nd　nf　nk]；其它参数可参考 rarx 。

4. roe 命令　求输出误差模型的递推估计。

格式：[thm, yhat]= roe(z, nn, adm, adg)

式中， z =[y　u]； nn =[nb　nf　nk]；其它参数可参考 rarx 。

5. rpem 命令　求通用参数模型的递推估计（递推预测误差方法）。

格式：[thm, yhat]= rpem(z, nn, adm, adg)

式中， z =[y　u]； nn =[na　nb　nc　nd　nf　nk)；其它参数可参考 rarx 。

6. rplr 命令　求通用参数模型的递推估计（递推伪线性回

归方法）。

格式：[thm, yhat]＝rplr(z, nn, adm, adg)

式中，z＝[y u]；nn＝[na nb nc nd nf nk]；其它参数可参考 rarx。

7. segment 命令 进行数据分段并跟踪变化剧烈的系统。

格式：[segm, v]＝segment(z, nn, r2, q)

式中，z＝[y u]；nn＝[na nb nk]（ARX 模型）；r2 为方程误差方差；q 为系统在采样点处跳跃变化的概率；segm 为分段数据的参数；v 为对应 segm 的损失函数。

10.2.7 模型结构处理

进行模型结构处理时可采用 fixpar、sett、thinit、unfixpar 等命令。

1. fixpar 命令 更新状态空间或 ARX 模型结构中的确定参数值。

格式：th = fixpar(thold, mat, ele, parval)

式中，th 和 thold 分别为新、旧 theta 矩阵；mat 为待处理的矩阵（若为状态空间模型，可为'A'、'B'、'C'、'D'、'K'、'x0' 中的一个；若为 ARX 模型，则可为'A1'、'A2'、'B0'、'B1'等）；ele 为待处理的矩阵元素序号 $n\times 2$ 维矩阵，其每行的两个值分别对应待处理矩阵元素的行、列号；parval 为更新的参数值 n 维矢量，与 ele 的指示相对应。

2. sett 命令 设置 theta 格式中的采样间隔。

格式：thn = sett(th, T)

Gn = sett(G, T)

式中，th 为原 theta 模型；thn 为更新 T 后的 theta 模型；G 为原频率函数，Gn 为更新 T 后的频率函数。

3. thinit 命令 在保证预测器或系统稳定性的基础上给 theta 参数赋随机初始值。

格式：th = thinit(tho)

th = thinit(tho, r, par, sp)

式中， tho 为原 theta 模型； r 为随机初始参数的方差； par 为初始参数的均值； sp 为稳定性要求， sp ='p' 时，要保证预测器的稳定性， sp ='s' 时，要保证系统的稳定性， sp ='b' 时要保证两者的稳定性。

4. unfixpar 命令 更新状态空间模型或 ARX 模型结构中的待估计参数。

格式： th = unfixpar(thold, mat, ele)

式中，各参数含义可参考 fixpar 。

10.2.8 模型表达

进行模型表达时可采用bodeplot、ffplot、idplot、nyqplot、present 、 zpplot 等命令。

1. bodeplot 命令 绘出传递函数或频谱的伯德图。

格式： bodeplot(G)

bodeplot(G, sd)

式中， G 为由 th2ff 、 spa 、 etfe 等命令产生的标准频率函数格式； sd 为标准差。

2. ffplot 命令 绘出传递函数或频谱的频率特性图。

格式： ffplot(G)

ffplot(G, sd)

与 bodeplot 的不同之处为 ffplot 绘图时为线性频率刻度，频率单位为 Hz，而 bodeplot 则为对数频率刻度，频率单位为 rad/s 。

3. idplot 命令 绘出输入-输出关系图。

格式： idplot(z)

idplot(z, int)

式中， z =[y u]; 若有 int 矢量，则使用 int 所给出的点的数据绘图，否则使用所有的数据。

4. nyqplot 命令 绘出传递函数的奈奎斯特图。

格式： nyqplot(G)

nyqplot(G, sd)

式中，G为th2ff、spa、etfe等命令产生的标准频率函数格式；sd为标准差。

5. present 命令 在屏幕上显示参数模型。

格式：present(th)

6. zppolt 命令 绘出零极点图。

格式：zpplot(zepo)

zpplot(zepo, sd)

zpplot(zepo, mode)

zpplot(zepo, sd, mode)

绘图用'o'表示零点，用'×'表示极点；sd 为标准差；mode 为绘图方式，当有多个输入时，若 mode ='sub'，则一屏中显示多个图（缺省），若 mode ='same'，所有图都绘制在一个大图中，用回车键切换，若 mode ='dep'，在绘制下一个图时，除去前一个图。

10.2.9 信息提取

进行信息提取时可采用 getmfth、getncap、getff、gett、getzp 等命令。

1. getmfth 命令 可得到定义模型结构的 m 文件名。

格式：mfname = getmfth(th)

2. getncap 命令 可得到模型的数据点数和参数个数。

格式：[Ncap, D]= getncap(th)

式中，Ncap为用来估计th的数据点数；D为模型结构中估计参数的个数。

3. getff 命令 选择用于绘图的频率函数。

格式：[w, amp, ph]= getff(G, nu, ny)

式中，w 为以 rad/s 为单位的频率刻度；amp 为幅频特性；ph 为相频特性；G 为频率函数格式；nu 和 ny 分别为输入和输出的维数。若想得到幅值标准差和相位标准差，则可用如下命令格

式：

[w, amp, ph, sd-amp, sd-ph]= getff(G, nu, ny)。

4. gett 命令 可得到采样间隔。

格式：T = gett(th)

5. getzp 命令 可得到 zepo 格式中的零极点。

格式：[z, p]= getzp(zepo, ku, ky)

式中，z、p 分别为零点和极点；ku 和 ky 分别为输入和输出序号。

10.2.10 模型结构选择

选择模型结构时可采用 arxstruc、ivstruc、selstruc、struc 等命令。

1. arxstruc 命令 计算 ARX 类模型的损失函数。

格式：v = arxstruc(z, zv, nn)

式中，z =[y u]；nn 是 ARX 结构参数；zv 是进行校验的输入输出数据；矢量 v 的第一行是对应 nn 所定义结构的损失函数。

2. ivstruc 命令 计算单输出 ARX 类模型的输出误差函数。

格式：v = ivstruc(ze, zv, nn)

式中，ze =[y u]；nn 是 ARX 结构参数；zv 是进行校验的输入输出数据；矢量 v 的第一行是对应 nn 所定义结构的输出误差函数。

3. selstruc 命令 根据各种准则选择模型结构。

格式：nn = selstruc(v, c)

[nn, wm]= selstruc(v, c)

式中，v 包含不同结构信息的矩阵，一般是由 arxstruc 或 ivstruc 命令得到的；c 是判据选择。若 c ='PLOT'，则绘出损失函数图和作为估计参数个数的函数的条件数；若 c ='AIC'，则根据 Akaike 信息理论准则自动选择结构；若 c ='MDL'，则用 Rissanen 最小长度准则；若 c 是一个数值，则通过最小化下列目标函数来选择结构：

$$J=(1+C\,d/N)V_{\mathrm{d}}$$

式中，d 为估计参数的个数；N 为数据个数；V_{d} 为相应模型的损失函数。

4. struc 命令 求用于 arxstruc 和 ivstruc 命令的典型结构矩阵。

格式：NN = struc(NA, NB, NK)

式中，NA、NB、NK 分别为包含要测试的 na, nb, nk 阶数的矩阵。

10.2.11 模型不确定性评估和模型校验

进行模型不确定性评估和模型校验时可采用 idsimsd、compare、idsim、pe、predict、resid 等命令。

1. idsimsd 命令 通过模型响应仿真说明不确定性。

格式：idsimsd(v, th)

idsimsd(v, th, N)

式中，v 是输入矢量，th 是 theta 格式模型。建立 10 个与 th 的协方差信息一致的随机模型，并将这些模型对 v 的响应绘制在同一图中，若参数中有 N，则为 N 个随机模型。

2. compare 命令 将仿真的或预测的输出与测量输出相比较。

格式：yh = compare(z, th, m)

式中，z =[y u]；m 为预测步数，m =∞时为纯仿真(缺省)；yh 为仿真输出。此命令还可绘出 yh 和 y 的曲线，并显示这两个信号的均方拟合情况（红实线是 yh，绿实线是 y）。

3. idsim 命令 仿真给定系统。

格式：y = idsim(z, th)

式中，z =[u e]；th 为 theta 模型。

4. pe 命令 计算预测误差。

格式：e = pe(z, th)

式中，e 为预测误差；z =[y u]；th 为 theta 模型。

更完整的格式为：

[e, v, w, A, B, C, D, F]＝ pe[z, th]

式中，A、B、C、D、F为通用参数中的多项式；w＝(B/F)u；v ＝ A[y−w]。

5. predict 命令　计算 m 步预测。

格式：yp ＝ predict(z, th, m)

式中，z ＝[y　u]；th 为 theta 模型；m 为预测步数。

6. resid 命令　计算和测试与模型相关的残差。

格式：E ＝ resid(z, th)

式中，z ＝[y　u]；th 为 theta 模型。

参考文献

1 Natick, Mass. MATLAB User's Guide Math Works,Inc. 1990.

2 Ogata K. Discrete-time Control Systems.Englewood Cliffs, N.J.,:Prentice Hall,1987.

3 Ogata K. Modern Control Engineering, 2nd ed..Englewood Cliffs, N.J., :Prentice Hall,1990.

4. Ogata K. Solving Control Systems Problems with MATLAb. Englewood Cliffs,N.J.,:Prentice Hall,1994.



《电气自动化新技术丛书》

目　录

- 大功率交-交变频调速及矢量控制技术(第2版)　马小亮著　已出版
- 可编程序控制器及其应用　宣练中、王燕生等编　已出版
- 电气传动的脉宽调制控制技术　吴守箴　臧英杰著　已出版
- 智能控制系统及其应用　王顺晃　舒迪前编著　已出版
- 异步电动机直接转矩控制　李夙编　已出版
- 模糊控制原理与应用　诸静等著　已出版
- 开关型磁阻电动机调速控制技术　王宏华编著　已出版
- 滑模变结构控制　王丰尧编著　已出版
- 系统最优化及控制　符曦编著　已出版
- 通用变频器及其应用　满永奎、韩安荣、吴成东编著　已出版
- 计算机辅助设计技术及应用　杨竞衡主编　已出版
- 电力电子场控器件及其应用　张立、黄两一等编著　已出版
- 直流无刷电动机原理及应用　张琛编著　已出版
- 预测控制系统及其应用　舒迪前编著　已出版
- 现代计算机数控系统　冯勇等编著　已出版
- 同步电动机调速系统　李志民、张遇杰编著　已出版
- 现代矿井提升机电控系统　王清灵、龚幼民编著　已出版
- 电力电子器件及其应用　李序葆、赵永健编著　已出版
- 交流步进传动系统　孙鹤旭著　已出版
- 无速度传感器矢量控制原理与实践　冯垛生、曾岳南编著　已出版
- 机器人控制技术　孙迪生、王炎编著　已出版
- 带钢热连轧计算机控制　刘玠、孙一康等编著　已出版

- 执行电动机　王季秩、曲家骐编著　已出版
- MATLAB 语言与自动控制系统　魏克新、王云亮、陈志敏编著　已出版
- 电机控制专用集成电路　谭建成主编　已出版
- 电控及自动化设备可靠性工程技术　徐平、李全灿编著　已出版
- 神经元网络控制　王永骥、涂健编著　预计 1998 年 1 月出版
- 控制系统的故障诊断和容错控制　闻新、张洪钺、周露编著　预计 1998 年 1 月出版
- SPWM 变频应用技术　张燕宾编著　预计 1998 年 1 月出版
- 伺服控制系统中的传感器　曲家骐、王季秩编著　预计 1998 年 4 月出版
- 交流伺服系统原理及应用　徐殿国、高波、王炎编著　预计 1998 年 3 月出版
- 交流调速系统　陈伯时、陈敏逊编著　预计 1998 年 4 月出版
- 现代电力电子器件应用导则　王正元编著　预计 1998 年 4 月出版
- 直流调速系统全数字控制　叶王、皮壮行、马济泉、余锡仁编著　预计 1998 年 3 月出版
- 工业计算机网络与多媒体技术　张浩、张民编著　预计 1998 年 4 月出版
- 谐波抑制和无功功率补偿　王兆安、杨君、刘进军编著　预计 1998 年 4 月出版
- 交流电机数字控制系统　李永东、王长江编著　预计 1998 年 4 月出版
- 交、直流传动系统的自适应控制　夏超英编著　预计 1998 年 4 月出版
- 新型 PID 控制及其应用　陶永华编著　预计 1998 年 4 月出版
- 电力牵引交流传动与控制技术　黄济荣编著　预计 1998 年 8 月出版

注:电气自动化新技术丛书由机械工业出版社出版，新华书店发行、经售。在新华书店购不到时，可与中国自动化学会电气自动化专业委员会邮购部(300180 天津市津塘路 174 号 电话: 022-24962354)联系购买。